

CONFIDENTIAL
ATARI
PRELIMINARY

MONITOR

2 of 12

Send to Gary

TITLE "MONITOR ***** MONITP.SRC ***** 3/9/79 ***** 4:00:00 P.M."

CONSTANT EQUATES

0009	PUTTXT	=	\$9	"PUT TEXT RECORD" CIO COMMAND CODE
0007	GETCAR	=	\$7	"GET CHARACTER" CIO COMMAND CODE
0000	PUTCAR	=	\$8	"PUT CHARACTER" CIO COMMAND CODE
0000	INIMLL	=	\$00	INITIAL MEM LO LOW BYTE
0007	INIMLH	=	\$07	INITIAL MEM LO HIGH BYTE
009B	CR	=	\$9B	ASCII CARRIAGE RETURN
0001	GOOD	=	\$1	GOOD STATUS CODE
0057	WRITE	=	\$57	WRITE COMMAND
0052	READ	=	\$52	READ COMMAND
0053	STATC	=	\$53	STATUS COMMAND
0000	SEX	=	\$0	SCREEN EDITOR IOCB INDEX
007D	CLS	=	\$7D	CLEAR SCREEN CODE
0092	CTRLC	=	\$92	KEYBOARD CODE FOR 'CONTROL C'
008B	EOF	=	\$8B	CASSETTE END OF FILE CODE
0000	LIR0	=	\$0	LONG IRO TYPE CODE

0004	BUFFH	=	CASBUF+3/256	
0000	BUFFL	=	-256+BUFFH+CASBUF+3	BUFFER POINTER

THE FOLLOWING EQUATES ARE IN THE CARTRIDGE ADDRESS SPACE.

"B" CARTRIDGE ADDR'S ARE 8000-9FFF (36K CONFIG. ONLY)

"A" CART. ADDR'S ARE A000-BFFF (36K CONFIG. ONLY)

"A" CART. ADDR'S ARE 8000-BFFF (48K CONFIG. ONLY)

00FA	CARTCB	==	00FA	CARTRIDGE COLD START ADDRESS
00FC	CART	==	00FC	CARTRIDGE AVAILABLE FLAG BYTE
00FD	CARTFO	==	00FD	CARTRIDGE FLAG BYTE. BIT 0=FLAG1.
00FE	CARTAD	==	00FE	2-BYTE CARTRIDGE START VECTOR

CARTRIDGE FLAG ACTION DEFINITIONS

BIT	ACTION IF SET
7	SPECIAL -- DON'T POWER-UP, JUST RUN CARTRIDGE
6-3	NONE
2	RUN CARTRIDGE
1	NONE
0	BOOT DOS

NOTE

1. IF BIT2 IS 0, GOTO BLACKBOARD MODE.

2. IF BIT0 SET, THE DISK WILL BE BOOTED BEFORE ANY
OTHER ACTION.

POWER-UP VECTOR

* = \$FFFC

PVECT WORD PWRUP POWER-UP VECTOR

ENTRY POINT VECTOR

* = BLACKBV

I-471 4C 23 F2

JMP BIGNON

BLACK BOARD VECTOR

* = WARMBV

I-474 4C 06 90

JMP \$9006

WARM START VECTOR

* = COLD8V

I-477 4C 00 90

JMP \$9000

COLD START VECTOR (\$9000 FOR RAM VECTOR WRITING)

* = \$9000

7000 20 0C 90

JBR \$900C

7003 4C 23 F1

JMP PWRUP

(TO HANDLE RAM VECTOR WRITING)

7006 20 0C 90

JBR \$900C

9009 4C 1B F1

JMP REBET

* = MONOR0

HANDLER TABLE ENTRIES

I-0E3 50

TBLENT

BYTE "P"

I-0E4 30 E4

WORD PRINTV

I-0E4 43

BYTE "C"

I-0E7 40 E4

WORD CABETV

I-0E9 45

BYTE "E"

I-0EA 00 E4

WORD EDITRV

I-0EC 53

BYTE "S"

I-0ED 10 E4

WORD SCREENV

I-0EF 4B BYTE "K"
I-0F0 20 E4 WORD KEYBDV

000E TBLLEN = IDENT-TBLENT-1 HANDLER TABLE LENGTH.

***** PRINT MESSAGES *****

I-0F2 7D 41 54 IDENT BYTE CLS, "ATARI COMPUTER - MEMO PAD", CR
I-0F5 41 52 49

I-0FB 20 43 4F
I-0FB 4D 80 93
I-0FE 54 46 52
I-101 20 2D 2D
I-104 4D 45 4D
I-107 4F 2D 5D
I-10A 41 44 9B

00F0 IDENTH = IDENT/256
00F2 IDENTL = -256*IDENTH+IDENT SYSTEM I.D. MSQ POINTER

I-10D 42 4F 4F DERR BYTE "BOOT ERROR", CR

F110 54 2D 45
F113 52 52 4F
F116 52 9B

00F1 DERRH = DERR/256
00D0 DERRL = -256*DERRH+DERR DISK ERROR MSQ POINTER

DEVICE/FILENAME SPECIFICATIONS

F11B 45 3A 9B OPNEDT BYTE "B", CR "OPEN SCREEN EDITOR" DEVICE SPEC

00F1 OPNH = OPNEDT/256
001B OPNL = -256*OPNH+OPNEDT SCREEN EDITOR OPEN POINTER

***** RESET BUTTON ROUTINE STARTS HERE *****

F11B 7B	RESET	BEI	DISABLE IRQ INTERRUPTS
F11C AD 44 03		LDA	COLDST
			WERE WE IN MIDDLE OF COLDSTART?
I-11F D0 04		BNE	PWRUP
			YES, GO TRY IT AGAIN
I-121 A9 FF		LDA	#FF
			SET WARM START FLAG
I-123 D0 03		BNE	PWRUP1

***** POWER UP ROUTINES START HERE *****


```

F125 78      PWRUP  BEI      DISABLE IRQ INTERRUPTS
F126 A9 00    LDA      #0      CLEAR WARMSTART FLAG
F128 B5 08    PWRUP1 STA      WARMST
F12A D8      CLD      CLEAR DECIMAL FLAG
F12B A2 FF    LDX      #FF
F12D 9A      TXB
F12E 20 3F F2 JSR      SPECL    SET STACK POINTER
F131 20 81 F2 JSR      HARDI    CARTRIDGE SPECIAL CASE?
F134 A5 08    LDA      WARMST    DO HARDWARE INITIALIZATION
F136 D0 28    BNE      ZOBGRAM   IS IT WARMSTART?
                                      YES, ONLY ZERO OS RAM

```

```

F138 A9 00    ZERDRM LDA      #0
F13A A0 08    LDY      #WARMST
F13C B5 04    STA      RAMLO
F13E B5 03    STA      RAMLO+1   INITIALIZE RAM POINTER
F140 91 04    CLRRAM STA      (RAMLO),Y  CLEAR MEMORY LOC.
F142 C8      INY
F143 C0 00    CPY      #0      AT END OF PAGE?
F145 D0 F9    BNE      CLRRAM
F147 E6 03    INC      RAMLO+1   YES, INCR PAGE POINTER
F149 A6 03    LDX      RAMLO+1
F14B E4 06    CPX      TRAMBZ
F14D D0 F1    BNE      CLRRAM    AT END OF MEM?
                                      NO

```

```

F14F AD 72 E4 INITIALIZE DOBVEC TO POINT TO SIGNON (BLACKBOARD)
F152 B5 0A    LDA      BLKBDV+1
F154 AD 73 E4 STA      DOBVEC    USE BLACKBOARD VECTOR
F157 B5 08    LDA      BLKBDV+2    FOR DOBVEC
F159 A9 FF    STA      DOBVEC+1
F15B BD 44 02 LDA      #FF
F15E D0 13    STA      COLDST    SET TO SHOW IN MIDDLE OF COLDSTART
F15E D0 13    BNE      ESTBCH    GO AROUND ZOBGRAM

```

```

F160 A2 00    ZOBGRAM LDX      #0      CLEAR OS RAM (FOR WARMSTART)
F162 BA      TXA
F163 9D 00 02 ZOSRM2 STA      #200,X  CLEAR PAGES 2 AND 3
F166 9D 00 03 STA      #300,X
F169 CA      DEX
F16A D0 F7    BNE      ZOSRM2
F16C A2 10    LDX      #INTZBB
F16E 93 00    ZOSRM3 STA      0,X    CLEAR ZERO PAGE LOCATIONS INTZBB-7F
F170 E8      INX
F171 10 FB    BPL      ZOSRM3

```

```

F173 A9 02    ESTBCH LDA      #LEDGE  ESTABLISH SCREEN MARGINS
F175 B5 52    STA      LMRGN
F177 A9 27    LDA      #REDGE
F179 B5 53    STA      RMRGN

```

```

F17B A2 23    OPBYB LDX      #23      MOVE VECTOR TABLE FROM ROM TO RAM
F17D BD 80 E4 MOVVEC LDA      VCTABL,X  ROM TABLE
F180 9D 00 02 STA      INTABB,X    TO RAM
F183 CA      DEX
F184 10 F7    BPL      MOVVEC
F186 20 94 F2 JSR      OSRAM    DO O.B. RAM SETUP
F189 5B      CLI      ENABLE IRQ INTERRUPTS

```

LINK HANDLERS

F18A A2 0E	LDX	#TBLEN	
F18C BD E3 F0	NXTENT	LDA	TBLEN, X
F18F 9D 1A 03		STA	HATABB, X
F192 CA	DEX		
F193 10 F7	BPL	NXTENT	DONE WITH ALL ENTRIES?

INTERROGATE CARTRIDGE ADDR. SPACE TO SEE WHICH CARTRIDGES THERE ARE

F195 A2 00	LDX	#0	
F197 B6 07	BTX	TBTDAT	CLEAR "B" CART. FLAG
F199 B6 06	BTX	TRAMSZ	CLEAR "A" CART. FLAG
F19B AE E4 02	LDX	RAMSIZ	
F19E E0 90	CPX	#90	RAM IN "B" CART. SLOT?
F1A0 B0 0A	BCS	ENDBCK	
F1A2 AD FC 9F	LDA	CART-\$2000	NO,
F1A5 D0 05	BNE	ENDBCK	CART. PLUGGED INTO "B" SLOT?
F1A7 E6 07	INC	TBTDAT	YES, SET "B" CART. FLAG
F1A9 20 3C F2	JBR	CBINI	INITIALIZE CARTRIDGE "B"
F1AC AE E4 02	ENDBCK	LDX	RAMSIZ
F1AF E0 B0	CPX	#B0	RAM IN "A" CART. SLOT?
F1B1 B0 0A	BCB	ENDACK	
F1B3 AE FC BF	LDX	CART	NO,
F1B6 D0 05	BNE	ENDACK	CART. PLUGGED INTO "A" SLOT?
F1B8 E6 06	INC	TRAMSZ	YES, SET "A" CART. FLAG
F1BA 20 39 F2	JBR	CAINI	INITIALIZE CARTRIDGE "A"

OPEN SCREEN EDITOR

F1BD A9 03	ENDACK	LDA	#3
F1BF A2 00	LDX	#SEX	
F1C1 9D 42 03	STA	ICCOM, X	OPEN I/O COMMAND
F1C4 A9 18	LDA	#OPNL	
F1C6 9D 44 03	STA	ICBAL, X	
F1C9 A9 F1	LDA	#OPNI	
F1CB 9D 45 03	STA	ICBAH, X	SET BUFFER POINTER TO OPEN SCREEN EDITOR
F1CE A9 0C	LDA	#C	
F1D0 9D 4A 03	STA	ICAX1, X	SET UP OPEN FOR INPUT/OUTPUT
F1D3 20 56 E4	JBR	CIOV	GO TO CIO
F1D6 10 03	BPL	SCRNOK	BR IF NO ERROR
F1DB 4C 25 F1	JMP	PWRUP	RETRY PWRUP IF ERROR (SHOULD NEVER HAPPEN!)
F1DB E8	SCRNOK	INX	SCREEN OK, SO WAIT FOR VBLANK TO
F1DC D0 FD	BNE	SCRNOK	BRING UP THE DISPLAY
F1DE C8	INX		
F1DF 10 FA	BPL	SCRNOK	

DO CASSETTE BOOT

F1E1 20 B2 F3	JBR	CSBOOT	CHECK, BOOT, AND INIT
---------------	-----	--------	-----------------------

```

; CHECK TO SEE IF EITHER CARTRIDGE WANTS DISK BOOT
F1E4 A5 06 LDA TRAMBZ CHECK BOTH CARTRIDGES
F1E6 05 07 ORA TSTDAT
F1E8 F0 12 BEQ NOCART NEITHER CARTRIDGE LIVES
F1EA A5 06 LDA TRAMBZ "A" CART?
F1EC F0 03 BEQ NOA1 NO
F1EE AD FD BF LDA CARTF0 GET CARTRIDGE MODE FLAG
F1F1 A6 07 NOA1 LDX TSTDAT "B" CART?
F1F3 F0 03 BEQ NOB1 NO
F1F5 0D FD 9F ORA CARTF0-$2000 ADD OTHER FLAG
F1FB 29 01 NOB1 AND #1 DOES EITHER CART WANT BOOT?
F1FA F0 03 BEQ NOBOOT NO

```

```

; DO DISK BOOT
F1FC 20 CF F2 NOCART JSR BOOT CHECK, BOOT, AND INIT

```

```

; GO TO ONE OF THE CARTRIDGES IF THEY SO DESIRE
F1FF A9 00 NOBOOT LDA #0
F201 BD 44 03 STA COLDST RESET TO SHOW DONE WITH COLDSTART
F204 A5 06 LDA TRAMBZ "A" CART?
F206 F0 0A BEQ NOA2 NO
F208 AD FD BF LDA CARTF0 GET CARTRIDGE MODE FLAG
F20B 29 04 AND #4 DOES IT WANT TO RUN?
F20D F0 03 BEQ NOA2 NO
F20F 6C FA BF JMP (CARTCB) RUN "A" CARTRIDGE
F212 A5 07 NOA2 LDA TSTDAT "B" CART?
F214 F0 0A BEQ NOCAR2 NO
F216 AD FD 9F LDA CARTF0-$2000 GET "B" MODE FLAG
F219 29 04 AND #4 DOES IT WANT TO RUN?
F21B F0 DF BEQ NOCART NO
F21D 6C FA 9F JMP (CARTCB-$2000) RUN "B" CARTRIDGE

```

```

; NO CARTRIDGES, OR NEITHER WANTS TO RUN,
; SO GO TO DOBVEC (DOS, CASSETTE, OR BLACKBOARD)
F220 6C 0A 00 NOCAR2 JMP (DOBVEC)

```

```

; PRINT SIGN-ON MESSAGE
F223 A2 F2 SIGNON LDX #IDENTL
F225 A0 F0 LDY #IDENTH
F227 20 B5 F3 JSR PUTLIN GO PUT SIGN-ON MSG ON SCREEN

```

```

; BLACKBOARD ROUTINE
F22A 20 30 F2 BLACKB JSR BLKB2 "JSR EQETCH"
F22D 4C 2A F2 JMP BLACKB FOREVER
F230 AD 05 E4 BLKB2 LDA EDITRV+5 HIGH BYTE
F233 4B PHA
F234 AD 04 E4 LDA EDITRV+4 LOW BYTE
F237 4B PHA
F238 60 RTS SIMULATES "JMP (EDITRV)"

```

```

; CARTRIDGE INITIALIZATION INDIRECT JUMPS
F239 6C FE BF CAINI JMP (CARTAD)
F23C 6C FE 9F CBINI JMP (CARTAD-$2000)

```

SUBROUTINES

CHECK FOR HOW MUCH RAM & SPECIAL CARTRIDGE CASE.
IF SPECIAL CARTRIDGE CASE, DON'T GO BACK -- GO TO CART.

ADDRESS	OPCODE	OPERAND	INSTR	COMMENT
I-23F	AD FC BF	SPECL	LDA	CART
F242	D0 12		BNE	ENSPEC
F244	EE FC BF		INC	CART
F247	AD FC BF		LDA	CART
I-24A	D0 0A		BNE	ENSPEC
I-24C	AD FD BF		LDA	CARTFO
I-24F	29 80		AND	##80
F251	F0 03		BEQ	ENSPEC
F253	4C FE BF		JMP	(CARTAD)

CHECK FOR RAM OR CART
GO IF NOTHING OR MAYBE RAM
NOW DO RAM CHECK
IS IT RAM?
NO
YES,
MASK OFF SPECIAL BIT
BIT BET?
YES, GO RUN CARTRIDGE

CHECK FOR AMOUNT OF RAM

ADDRESS	OPCODE	OPERAND	INSTR	COMMENT
F256	CE FC BF		ENSPEC	DEC
F259	A9 00		LDA	NO
F25B	85 05		BTA	RAMLO+1
F25D	A9 10		LDA	##10
F25F	85 06		BTA	TRAMBZ
I-261	A0 00		LDY	NO
F263	81 05		LDA	(RAMLO+1),Y
F265	85 07		BTA	TBTDAT
F267	49 FF		EOR	##FF
F269	85 04		BTA	RAMLO
F26B	91 05		STA	(RAMLO+1),Y
F26D	81 05		LDA	(RAMLO+1),Y
F26F	C9 04		CMF	RAMLO
F271	D0 00		BNE	ENDRAM
F273	A5 07		LDA	TBTDAT
F275	91 05		STA	(RAMLO+1),Y
F277	A5 06		LDA	TRAMBZ
F279	18		CLC	
F27A	89 10		ADC	##10

RESTORE RAM IF NEEDED

SET RAM POINTER TO 4K.

READ RAM LOCATION

SAVE DATA

INVERT IT.

SAVE INVERTED DATA

WRITE INVERTED DATA.

READ RAM AGAIN

IS IT THE INVERTED DATA?

YES.

RESTORE ORIGINAL RAM DATA

127C 85 06	STA	TRAMBZ	INCR. RAM POINTER BY 4K.
127E D0 E3	BNE	HOWMCH	GO FIND HOW MUCH RAM.
1280 60	ENDRAM	RTS	

HARDWARE INITIALIZATION

1281 A9 00	HARDI	LDA	#0
1283 AA	TAX		
1284 9D 00 D0	CLRCHP	STA	\$D000,X
1287 9D 00 D4		STA	\$D400,X
128A 9D 00 D2		STA	\$D200,X
128D 9D 00 D3		STA	\$D300,X
1290 E8	INX		
1291 D0 F1	BNE	CLRCHP	
1293 60	RTS		

0.5. RAM SETUP

1294 C6 11	DBRAM	DEC	BRKKEY	TURN OFF BREAK KEY FLAG
1296 A5 06	LDA	TRAMBZ		READ RAM SIZE IN TEMP REG
1298 8D E4 02	STA	RAMBIZ		SAVE IT IN RAM SIZE
129B 8D E6 02	STA	MEMTOP+1		INIT. MEMTOP ADDR HI BYTE
129E A9 00	LDA	#0		
12A0 8D E5 02	STA	MEMTOP		INIT. MEMTOP ADDR LO BYTE
12A3 A9 00	LDA	#INIML		
12A5 8D E7 02	STA	MEMLO		
12AB A9 07	LDA	#INIMH		
12AA 8D E8 02	STA	MEMLO+1		INITIALIZE MEMLO ADDR VECTOR
12AD 20 0C E4	JSR	EDITRV+8C		EDITOR INIT.
12B0 20 1C E4	JSR	SCREENV+8C		SCREEN INIT.
12B3 20 2C E4	JSR	KEYBDV+8C		KEYBOARD INIT.
12B6 20 3C E4	JSR	PRINTV+8C		PRINTER HANDLER INIT
12B9 20 4C E4	JSR	CASSETV+8C		CASSETTE HANDLER INIT
12BC 20 6E E4	JSR	CIOINV		CIO INIT.
12BF 20 65 E4	JSR	SIOINV		SIO INIT.
12C2 20 6B E4	JSR	INTINV		INTERRUPT HANDLER INIT.
12C5 AD 1F D0	LDA	CONBOL		
12C8 29 01	AND	#1		
12CA D0 02	BNE	NOKEY		GAME START KEY DEPRESSED?
12CC E6 4A	INC	CKEY		YES, SET KEY FLAG.
12CE 60	NOKEY	RTS		

DO BOOT OF DISK

12CF A5 08	BOOT	LDA	WARMST	
12D1 F0 0A		BEG	NOWARM	WARM START?
12D3 A5 09		LDA	BOOT?	YES,
12D5 29 01		AND	#1	
12D7 F0 03		BEG	NOINIT	VALID BOOT?
12D9 20 7E F3		JSR	DINI	YES, RE-INIT. DOS SOFTWARE
12DC 60	NOINIT	RTS		
12DD A9 01	NOWARM	LDA	#1	
12DF 8D 01 03		STA	DUNIT	ASSIGN DISK DRIVE NO.
12E2 A9 53		LDA	WSTATC	

12E4 8D 02 03	STA	DCOMND	SET UP STATUS COMMAND
12E7 20 53 E4	JSR	DSKINV	GO DO DISK STATUS
12EA 10 01	BPL	DOBOOT	IS STATUS FROM SIO GOOD?
12EC 60	RTS		NO, GO BACK WITH BAD BOOT STATUS
12ED A9 00	DOBOOT	LDA	#0
12EF 8D 0B 03	STA	DAUX2	
12F2 A9 01	LDA	#1	
12F4 8D 0A 03	STA	DAUX1	SET SECTOR # TO 1.
12F7 A9 00	LDA	#BUFFL	
12F9 8D 04 03	STA	DBUFFLQ	
12FC A9 04	LDA	#BUFFH	
12FE 8D 05 03	STA	DBUFFH	SET UP BUFFER ADDR
1301 20 9D F3	SECT1	JSR	GETSEC
1304 10 08	BPL	ALLSEC	GET SECTOR
1306 20 81 F3	BADDSK	JSR	STATUS O.K. ?
1309 A5 4B	LDA	DSKRDE	NO, GO PRINT DISK READ ERROR
130B F0 E0	BEQ	CASSBT	
130D 60	DOBOOT	RTS	CASSETTE BOOT?
130E A2 03	ALLSEC	LDX	YES, QUIT
1310 8D 00 04	RDBYTE	LDA	#3
1313 9D 40 02	STA	CASBUF+3, X	READ A BUFFER BYTE
1316 CA	DEX	DFAQB, X	STORE IT
1317 10 F7	BPL	RDBYTE	
1319 AD 42 02	LDA	DOBTAD	DONE WITH 4 BYTE TRANSFER ?
131C 85 04	STA	RAMLO	YES.
131E AD 43 02	LDA	BOOTAD+1	
1321 85 05	STA	RAMLO+1	PUT BOOT ADDR INTO 2. PAGE RAM
1323 AD 04 04	LDA	CASBUF+7	
1326 85 00	STA	DOSINI	ESTABLISH DOS INIT ADDRESS
1328 AD 05 04	LDA	CASBUF+8	
132B 85 00	STA	DOSINI+1	
132D A0 7F	MVBUFF	LDY	##7F
132F 89 00 04	MVNXB	LDA	YES, SET BYTE COUNT
1332 91 04	STA	CASBUF+3, Y	
1334 88	DEY	(RAMLO), Y	MOVE A BYTE FROM SECTOR BUFFER TO BOOT ADDR.
1335 10 F8	BPL	MVNXB	
1337 18	CLC		DONE ?
1338 A5 04	LDA	RAMLO	YES.
133A 69 80	ADC	##80	
133C 85 04	STA	RAMLO	
133E A5 05	LDA	RAMLO+1	
1340 69 00	ADC	#0	
1342 85 05	STA	RAMLO+1	INCR BOOT LOADER BUFFER POINTER.
1344 CE 41 02	DEC	DBSECT	
1347 F0 11	BEQ	ENBOOT	DECR # OF SECTORS.
1349 EE 0A 03	INC	DAUX1	MORE SECTORS ?
134C 20 9D F3	SECTX	JSR	YES, INCR SECTOR #
134F 10 DC	BPL	GETSEC	GO GET SECTOR
1351 20 81 F3	JSR	MVBUFF	STATUS O.K. ?
1354 A5 4B	LDA	DSKRDE	NO, GO PRINT DISK READ ERROR
1356 D0 AE	BNE	CASSBT	
1358 F0 F2	BEQ	BADDSK	IF CASSETTE, QUIT.
135A A5 4B	ENBOOT	SECTX	IF DISK, TRY SECTOR AGAIN.
135C F0 03	LDA	CASSBT	
135E 20 9D F3	JSR	XBOOT	CASSETTE BOOT ?
1361 20 6C F3	XBOOT	JSR	YES, GET EOF RECORD, BUT DON'T USE IT.
1364 80 A0	BCS	BLOAD	GO EXECUTE BOOT LOADER
1366 20 7E F3	JSR	BADDSK	IF BAD BOOT, DO IT OVER AGAIN
1369 E6 09	INC	DINI	GO INIT. SOFTWARE
		BOOT?	SHOW BOOT SUCCESS


```

F36B 60          RTB
F36C 1B          BLOAD CLC
F36D AD 42 02    LDA BOOTAD
F370 69 06       ADC #6
F372 85 04       STA RAMLO
F374 AD 43 02    LDA BOOTAD+1
F377 69 00       ADC #0
F379 85 05       STA RAMLO+1    PUT START ADDR OF BOOTLOADER INTO RAM
F37B 6C 04 00    JMP (RAMLO)
F37E 6C 0C 00    DINI JMP (DOSINI)

```

DISPLAY DISK READ ERROR MSG

```

F381 A2 0D       DSKRDE LDX #DERRL
F383 A0 F1       LDY #DERRH

```

PUT LINE ON SCREEN AT PRESENT CURSOR POSITION

X-REG -- LO BYTE, BEGIN ADDR OF LINE
Y-REG -- HI BYTE, BEGIN ADDR OF LINE

```

F385 8A          PUTLIN TXA
F386 A2 00       LDX #SEX
F388 9D 44 03    STA ICBAL,X
F38B 98          TVA
F38C 9D 45 03    STA ICBAL,X    SET UP ADDR OF BEGIN OF LINE
F38F A9 09       LDA #PUTTXT
F391 9D 42 03    STA ICCOM,X    "PUT TEXT RECORD" COMMAND
F394 A9 FF       LDA #6FF
F396 9D 48 03    STA ICBLL,X    SET BUFFER LENGTH
F399 20 56 E4    JSR CIOV    PUT LINE ON SCREEN
F39C 60          RTS

```

GET SECTOR FROM DISK 0

```

F39D A5 1B       GETSEC LDA CASSET
F39F F0 03       BEQ DISKM    CASSETTE BOOT ?
F3A1 4C 7A E4    JMP RBLOKV    YES, GO TO READ BLOCK ROUTINE
F3A4 A9 52       DISKM LDA #READ
F3A6 8D 02 03    STA DCOMND    GET READ SECTOR COMMAND
F3A9 A9 01       LDA #1
F3AB 8D 01 03    STA DUNIT    GET DRIVE NO. TO DRIVE 0
F3AE 20 53 E4    JSR DSKINV    GET SECTOR
F3B1 60          RTS

```

DO CHECK FOR CASSETTE BOOT & IF SO, DO BOOT

```

F3B2 A5 08       CSBOOT LDA WARMST    WARMSTART?
F3B4 F0 0A       BEQ CSBOT2    NO
F3B6 A5 09       LDA BOOT?    GET BOOT FLAG

```

I-308 29 02	AND	#2	WAS CASSETTE BOOT SUCCESSFUL?
I-30A F0 03	BEG	NOC8B2	NO
I-30C 20 E1 F3	JSR	CINI	YES, INIT CASSETTE SOFTWARE
I-30F 60	NOC8B2	RTB	
I-3C0 A5 4A	C6BOT2	LDA	CKEY
I-3C2 F0 1C	BEG	NOC8BT	"C" KEY FLAG SET ?
I-3C4 A9 80	LDA	#80	YES,
I-3C6 85 3E	STA	FTYPE	SET LONG IRO TYPE
I-3C8 E6 4B	INC	CAS8BT	SET CASSETTE BOOT FLAG
I-3CA 20 7D E4	JSR	C8OP1V	OPEN CASSETTE FOR INPUT
I-3CD 20 01 F3	JSR	SECT1	DO BOOT & INIT
I-3D0 A9 00	LDA	#0	
I-3D2 85 4B	STA	CAS8BT	REBET CASSETTE BOOT FLAG
I-3D4 85 4A	STA	CKEY	CLEAR KEY FLAG
I-3D6 06 09	ASL	BOOT?	SHIFT BOOT FLAG (NOW=2 IF SUCCESS)
I-3D8 A5 0C	LDA	DOSINI	
I-3DA 85 02	STA	CASINI	MOVE INIT ADDRESS FOR CASSETTE
I-3DC A5 0D	LDA	DOSINI+1	
I-3DE 85 03	STA	CASINI+1	
I-3E0 60	NOC8BT	RTB	

I-3E1 6C 02 00 CINI JMP (CASINI) INIT CASSETTE

SPARE BYTE OR MODULE TOO LONG FLAG

I-3E4 CRNTPC = *

0014 00 MONBPR * - 014 ADDORG-CRNTPC MONITP TOO LONG

END

SYMBOL TABLE

ADDRESS	030E	ADDRESS	0064	APP	D800	ALLPRT	D208
ATTACH	F30E	ANTIC	D400	APPEND	0001	APPMHI	000E
ATTACH	02FB	ATAN	BE43	ATTRACT	004D	AUDC1	D201
AUDC2	D203	AUDC3	D205	AUDC4	D207	AUDCTL	D208
AUDF1	D200	AUDF2	D202	AUDF3	D204	AUDF4	D206
BADDDK	F306	BADIOC	0086	BADMOD	0091	BFENHI	0035
BEENLO	0034	BITMAK	006E	BLACKB	F22A	BLIM	028A
BLKDD	F230	BLKBDV	E471	BLOAD	F36C	BOOT	F2CF
BOOT?	0009	BOOTAD	0242	BOTSCR	02BF	BPTR	003D
BRKART	0080	BRKKEY	0011	BUFADR	0015	BUFCNT	0068
BUFFH	0004	BUFFL	0000	BUFRFL	003B	BUFRHI	0033
BUFRLO	0032	BUFBTR	006C	CAINI	F239	CART	BFFC
CARTAD	BFFE	CARTCS	BFFA	CARTFQ	BFFD	CASQUE	03FD
CASCTV	E440	CASFL0	030F	CASINI	0002	CASORQ	EF41
CASRTV	004B	CASSET	0043	CAUX1	023C	CAUX2	023D
CBADUI	02EE	CBAUDL	02EE	CBINI	F23C	CCOMND	023B
CDEVIC	023A	CDTHA1	0226	CDTHA2	0228	CDTHF3	022A
CDTHF4	022C	CDTHF5	022E	CDTHV1	021B	CDTHV2	021A
CDTHV3	021C	CDTHV4	021E	CDTHV5	0220	CH	02FC
CH	02F2	CHACT	02F3	CHACTL	D401	CHAR	02FA
CHIAM	02F4	CHBASE	D409	CHKERR	00BF	CHKENT	003B
CHKRCH	0031	CHORQ	E000	CINI	F3E1	CIOCHR	002F
CIOINV	E46E	CIOORQ	E4A6	CIOV	E4B6	CIX	00F2
CKEY	004A	CLOBE	000C	CLRCH	F2B4	CLRRAM	F140
CLSY	007D	COLAC	0072	COLBK	D016	COLCRS	0055
COLDET	0244	COLD8V	E477	COLINC	007A	COLOR0	02C4
COLR1	02C5	COLOR2	02C6	COLOR3	02C7	COLOR4	02C8
COLP10	D016	COLPF1	D017	COLPF2	D018	COLPF3	D019
COLPH0	D012	COLPH1	D013	COLPH2	D014	COLPH3	D015
COLRSH	004F	CONGOL	D01F	COS	B073	COUNTR	007E
CH	009B	CRETRY	0036	CRITIC	0042	CRNTPC	F3E4
CRGIND	02F0	CRSROR	008D	CSBOOT	F3B2	CSBOT2	F3C0
CRTIV	E47D	CSTAT	0288	CTIA	D000	CTRLC	0092
DAUX1	030A	DAUX2	030B	DBSECT	0241	DBUFHI	0305
DBUFLO	0304	DBYTH1	0309	DBYTLO	0308	DCB	0300
DCONND	0302	DDEVIC	0300	DEQFLO	00FB	DEQON	0004
DELETE	0021	DELTA0	0077	DELTAR	0076	DERR	F10D
DERR1	00F1	DERRL	000D	DERROR	0090	DFLA08	0240
DIGHT	00F1	DINDEX	0057	DINI	F37E	DIRECT	0002
DISK	0044	DISKIV	E450	DISKM	F3A4	DISPLY	0053
DLITTH	D403	DLITBL	D402	DMACTL	D400	DMASK	02A0
DMACK	008B	DMBOOT	F2E0	DMBINT	000C	DMBVEC	000A
DMAMN	0011	DRETRY	0027	DRAMSK	004E	DSKFMS	001B
DSKINV	E453	DSKORQ	EDEA	DSKRDE	F3B1	DSKTIM	0246
DSKUL	001A	DSPFLO	02FE	DSTAT	004C	DSTATB	0303
DTIR0	0306	DUNIT	0301	DUNUSE	0307	DVSTAT	02EA
EDITRV	E400	EEXP	00ED	ENBOOT	F35A	ENDACK	F1BD
ENDICK	F1AC	ENDPT	0074	ENDRAM	F2B0	ENSPEC	F256
EDF	008B	EDFERR	008B	ERRFLO	023F	ESCFLO	02A2
EXPON	00EF	ESTSCH	F173	EXP	DDC0	EXP10	DDCC
FADD	DA66	FASC	D8E6	FCHRFL	00F0	FDIV	DB2B
FILL	003F	FILDAT	02FD	FILFLO	02B7	FILLIN	0012
FLKOP	DDBD	FLDOR	DDB9	FLDIP	DD9C	FLDIR	DD98
FLPTH	00FC	FMQVE	DDB6	FMSZP0	0043	FMUL	DADB
FNCMIT	0072	FDRMAT	0022	FPI	D9D2	FPREC	0006
FPSCR	05E6	FPSCR1	05EC	FPTR2	00FE	FRO	00D4
FR	00E0	FR2	00E6	FRE	00DA	FREQ	0040
FRMRR	008C	FRX	00EC	FSCR	05E6	FSCR1	05EC
FSTOP	DDAB	FSTOR	DDA7	FSUB	DA60	FTYPE	003E

GETCAN	0007	GETCHR	0007	GETREC	0005	GETSEC	F39D
GLHAB	02E0	GOOD	0001	GPRIOR	026F	GRACIL	D01D
GRAPN	D011	GRAFFO	D00D	GRAFP1	D00E	GRAFP2	D00F
GRAPF1	D010	HARD1	F281	HATABS	031A	HITCLR	D01E
HOLD1	0051	HOLD2	029F	HOLD3	029D	HOLD4	02BC
HOLD5	02BD	HOLDCH	007C	HQWCH	F263	HPSMO	D004
HPSM1	D005	HPSM2	D006	HPSM3	D007	HPSPO	D000
HPSM4	D001	HPSM5	D002	HPSM6	D003	HSCROL	D404
ICAX1	034A	ICAX1Z	002A	ICAX2	034B	ICAX2Z	002B
ICBA1	0345	ICBAHZ	0025	ICBAL	0344	ICBALZ	0024
ICBL1	0349	ICBLHZ	0029	ICBL2	0348	ICBL2Z	0028
ICCOM1	0342	ICCOMT	0017	ICCOMZ	0022	ICDNO	0341
ICDM1Z	0021	ICID	0340	ICIDZ	0020	ICIDNO	002E
ICPT1	0347	ICPTHZ	0027	ICPTL	0346	ICPTLZ	0026
ICSPR	034C	ICSPRZ	002C	ICSTA	0343	ICSTAZ	0023
IDENT	F0F2	IDENTH	00F0	IDENTL	00F2	IFP	D9AA
INRUE	00F3	INIMLH	0007	INIMLL	0000	INSLR	0020
INODAT	007D	INTAB8	0200	INTEMP	022D	INTINV	E46B
INTIRO	E4D5	INTZB8	0010	INVFLQ	02B6	IDCB	0340
IOCHAB	0020	IOCB8Z	0010	IOCFRE	00FF	IRGEN	D20E
JRST	D20E	KBCODE	D209	KBD	0048	KBDORQ	F3E4
KEYINDV	E420	KEYDEL	02F1	LBFEND	05FF	LBPR1	057E
LBUFF	057F	LBUFF	0580	LEDGE	0002	LINBUF	0247
LINZIB	0000	LIRQ	0000	LMARQ	0052	LOCKFL	0023
LIRQ	DECD	LQ010	DED1	LQ0COL	0063	LQ0MAP	02B2
LPEN1	0234	LPENV	0235	MPFF	D000	MPPL	D00B
M1P1	D001	M1PL	D009	M2PF	D002	M2PL	D00A
M1P2	D003	M3PL	D008	MAXDEV	0021	MAXIOC	0080
M1P10	02E7	MENTOP	02E5	MLTTP	0066	MODEM	004D
MONIRQ	F0E3	MONBPR	0014	MOVVEC	F17D	MVBUFF	F32D
MVNXH	F32F	MXDMQD	0010	NEWCOL	0061	NEWROM	0060
NM1EN	D40E	NM1RES	D40F	NM1BT	D40F	NOA1	F1F1
NOA2	F212	NOB1	F1F8	NOBQOT	F1FF	NOCAR2	F220
NOCART	F1FC	NOCBKM	003C	NOCB82	F3BF	NOCB8T	F3E0
NOJNT1	F2DC	NOKEY	F2CE	NONDEV	0082	NOTE	0026
NOTICPN	00B5	NOHARN	F2DD	NSION	00EE	NVALID	00B4
NXTINT	F18C	OLDADR	005E	OLDCHR	005D	OLDCOL	005B
OLDHIM	005A	OPEN	0003	OPNEDT	F11B	OPNH	00F1
OPNIN	0004	OPNIND	000C	OPNL	0018	OPNOT	0008
OPNTH	0066	OPBYB	F17D	OSRAM	F294	OVRUN	008E
POPI	D004	POPL	D00C	PIPF	D005	P1PL	D00D
P3PF	D006	P3PL	D00E	P3PF	D007	P3PL	D00F
PACTL	D302	PADDL0	0270	PADDL1	0271	PADDL2	0272
PADDL3	0273	PADDL4	0274	PADDL5	0275	PADDL6	0276
PADDL7	0277	PBCTL	D303	PBPNT	001D	PBUF8Z	001E
PCOLR0	02C0	PCOLR1	02C1	PCOLR2	02C2	PCOLR3	02C3
PEN1	D40C	PENV	D40D	PIA	D300	PLYARQ	05E0
PLYEV	D440	PHBASE	D407	PPOINT	0025	POKEY	D200
POKMBK	0010	PORTA	D300	PORTB	D301	POT0	D200
POT1	D201	POT2	D202	POT3	D203	POT4	D204
POT5	D205	POT6	D206	POT7	D207	POT8	D208
PRINTH	0050	PRINTV	E430	PRIOR	D01B	PRNBUF	03C0
PRNIRQ	EE78	PRVOPN	0081	PTEMP	001F	PTIHOT	001C
PTRIQ0	027C	PTRIQ1	027D	PTRIQ2	027E	PTRIQ3	027F
PTRIQ4	0280	PTRIQ5	0281	PTRIQ6	0282	PTRIQ7	0283
PUTCAR	0008	PUTCHR	000B	PUTLIN	F385	PUTREC	0009
PUTXT	0009	PWRUP	F125	PWRUP1	F128	RADFLQ	00FB
RADIN	0000	RAMLO	0004	RAMSIZ	02E4	RAMTOP	006A
RANDUM	D20A	RBLOKV	E47A	RDBYTE	F310	RDONLY	0087
READ	0052	RECVDN	0039	REDGE	0027	RENAME	0020

REFRT	F11B	RMARGN	0053	ROWAC	0070	ROWCRS	0054
RMING	0079	RTCLCK	0012	SAVADR	0068	SAVID	0316
SAVNR	0058	SCREDT	0045	SCRENV	E410	SCRFLQ	0288
SCRNM	0093	SCRNOK	F1DB	SDLSTH	0231	SDLSTL	0230
SDMCT	022F	SECT1	F301	SECTX	F34C	SENDEV	E468
SERIN	D20D	SEROUT	D20D	SETVBV	E45C	SEX	0000
SIFANT	006F	SHFLCK	028E	SIGNON	F223	SIN	BD81
SIOINV	E463	SIOORQ	E944	SIOV	E459	SIZEM	000C
SIZIPO	D008	SIZEP1	D009	SIZEP2	D00A	SIZEP3	D00B
SKCTH	D20F	SKRES	D20A	SKSTAT	D20F	BOUNDR	0041
SLTCT1	000E	SPECL	F23F	SQR	BE01	SRTIMR	022B
SRFLAQ	02FF	SRKCTL	0232	STACKP	0318	STATC	0053
STAT1B	000D	BTATU8	0030	BTICK0	027B	BTICK1	0279
BTICK2	027A	BTICK3	027B	BTIMER	D209	STRIGQ	0284
STRIG1	0283	STRIG2	0286	STRIG3	0287	SUBTMP	029E
SUCCCB	0001	SWPFLQ	007B	SYSVBV	E45F	TABMAP	02A3
TDLIN1	F0E3	TDLLEN	000E	TEMP	023E	TEMP1	0312
TEMP2	0314	TEMP3	0315	TIMER1	030C	TIMER2	0310
TIMFLQ	0317	TIMOUT	008A	TINDEX	0293	TMPCHR	0050
TMPGCL	0289	TMPLEBT	02A1	TMPROW	0288	TMPX1	029C
TRANK2	0006	TRIGQ	D010	TRIG1	D011	TRIG2	D012
TRIG3	D013	TRNRCD	0089	TSTAT	0319	T8TDAT	0007
TXTCN	0291	TXTM8C	0294	TXTOLD	0296	TXTRQW	0290
UNLOCK	0024	USAREA	0480	VBREAK	0204	VCOUNT	040B
VCTANL	E480	VDELAY	001C	VBLEBT	0200	VECTBL	E400
VIMJRO	0216	VINTER	0204	VKEYBD	0208	VPRCED	0202
VSCRIN	D405	VSERIN	020A	VSEKDC	020E	VSEROR	020C
VTIMR1	0210	VTIMR2	0212	VTIMR4	0214	VVBLKD	0224
VWIKI	0222	WARMST	000B	WARMSV	E474	WMODE	0289
WRITE	0087	WROLY	0083	WSYNC	D40A	XBOOT	F361
XIIVBV	E462	XMTDON	003A	ZERORH	F13B	ZI0CB	0020
ZORRAM	F160	ZORRM2	F163	ZORRM3	F16E	ZTEMP1	00F5
ZTEMP3	00F9	ZTEMP4	00F7				

DISPLAY HANDLER -- 10-30-78 -- DISPLC

TITLE 'DISPLAY HANDLER -- 10-30-78 -- DISPLC'

HANDLER DEPENDENT EQUATES

(009D	CR	=	\$9B	SYSTEM EOL
(007D	CLRCOD	=	\$7D	CLEAR SCREEN ATASCII CODE
(009F	CNTLI	=	\$9F	POKEY KEY CODE FOR ^I
(006B	FRMADR	=	SAVADR	
(0066	TOADR	=	MLTTMP	

*=EDITRV

SCREEN EDITOR HANDLER ENTRY POINT

F400 FB F3	EDITOR	WORD	EOPEN-1	
F402 33 F6		WORD	RETUR1-1	(CLOSE)
F404 3D F6		WORD	EETCH-1	
F406 A3 F6		WORD	EOUTCH-1	
F408 33 F6		WORD	RETUR1-1	(STATUS)
F40A 3C F6		WORD	NOFUNC-1	(SPECIAL)
F40C 4C E4 F3		JMP	PWRONA	
F40F 00		BYTE	0	ROM FILLER BYTE

*=SCRENV

DISPLAY HANDLER ENTRY POINT

E410 F5 F3	DISPLA	WORD	DOPEN-1	
E412 33 F6		WORD	RETUR1-1	(CLOSE)
E414 92 F5		WORD	OETCH-1	
E416 B6 F5		WORD	OUTCH-1	
E418 33 F6		WORD	RETUR1-1	(STATUS)
E41A FB F6		WORD	DRAW-1	(SPECIAL)
E41C 4C E4 F3		JMP	PWRONA	
E41F 00		BYTE	0	ROM FILLER BYTE

*=KEYBDV

KEYBOARD HANDLER ENTRY POINT

F420 33 F6	KBD#ND	WORD	RETUR1-1	
F422 33 F6		WORD	RETUR1-1	(CLOSE)
F424 E1 F6		WORD	KOETCH-1	
F426 3C F6		WORD	NOFUNC-1	(OUTCH)
F428 33 F6		WORD	RETUR1-1	(STATUS)
F42A 3C F6		WORD	NOFUNC-1	(SPECIAL)
F42C 4C E4 F3		JMP	PWRONA	
F42F 00		BYTE	0	ROM FILLER BYTE

INTERRUPT VECTOR TABLE ENTRY

I:48B BE FF	*=VCTABL-INTABS+VKEYBD	WORD	PIRQ	KEYBOARD IRQ INTERRUPT VECTOR
-------------	------------------------	------	------	-------------------------------

DISPLAY HANDLER -- 10-30-78 -- DISPLC

==KBDDORQ

1-3E4 A9 FF	PWRONA	LDA	#FF	
1-3E6 BD FC 02		STA	CH	
1-3E9 AD E6 02		LDA	MEMTOP+1	
1-3EC 29 F0		AND	#F0	INSURE 4K PAGE BOUNDARY
1-3EE B5 6A		STA	RAMTOP	
1-3F0 A9 40		LDA	#40	DEFAULT TO UPPER CASE ALPHA AT PWRON
1-3F2 BD BE 02		STA	SHFLOK	
1-3F5 60		RTB		POWER ON COMPLETED

BEGIN DISPLAY HANDLER OPEN PROCESSING

I3F6 A5 2B	DOPEN	LDA	ICAX2Z	GET AUX 2 BYTE
F3FB 29 0F		AND	#0F	
I3FA D0 0B		BNE	OPNCOM	IF MODE ZERO, CLEAR ICAX1Z
I3EC A5 2A	EQPEN	LDA	ICAX1Z	CLEAR "CLR INHIBIT" AND "MXD MODE" BITS
I3FE 29 0F		AND	#0F	
I400 B5 2A		STA	ICAX1Z	
I402 A7 00		LDA	#0	
F404 B5 57	OPNCOM	STA	DINDEX	
I40A A9 E0		LDA	#0E0	INITIALIZE GLOBAL VBLANK RAM
F40B BD F4 02		STA	CHDAS	
I40B A9 02		LDA	#2	
I40D BD F3 02		STA	CHACT	
I410 BD 2F 02		STA	SDMCTL	TURN OFF DMA NEXT VBLANK
F413 A9 01		LDA	#SUCCESS	
F41B B9 AC		STA	DBSTAT	CLEAR STATUS
F417 A7 C0		LDA	#C0	DO IRGEN
I419 05 10		ORA	POKMSK	
I41B B5 10		STA	POKMSK	
I41D BD 0E D2		STA	IRGEN	
F420 A9 00		LDA	#0	
F422 BD 93 02		STA	TINDEX	TEXT INDEX MUST ALWAYS BE 0
F425 B5 64		STA	ADDRESS	
I427 B5 7B		STA	SWPFL0	
I429 BD F0 02		STA	CRSINH	TURN CURSOR ON AT OPEN
I42C A0 0E		LDY	#14	CLEAR TAB STOPS
F42E A9 D1		LDA	#1	INIT TAB STOPS TO EVERY 8 CHARACTERS
F430 99 A3 02	CLRTBS	STA	TABMAP	
F433 B8		DEY		
I434 10 FA		BPL	CLRTBS	
I436 A2 04		LDX	#4	LOAD COLOR REGISTERS
I43B BD C1 FE	DOPENB	LDA	COLRTB,X	
F43B 9D C4 02		STA	COLORO,X	
F43E CA		DEX		
F43F 10 F7		BPL	DOPENB	
I441 A4 6A		LDY	RAMTOP	DO TXTHSC=#2C40 (IF MEMTOP=3000)
I443 B8		DEY		
I444 BC 95 02		BTY	TXTHSC+1	
F447 A9 60		LDA	#60	
F449 BD 94 02		STA	TXTHSC	
F44C A5 57		LDX	DINDEX	
I44E BD 69 FE		LDA	ANCONV,X	CONVERT IT TO ANTIC CODE
I451 D0 04		BNE	DOPENA	IF ZERO, IT IS ILLEGAL
F453 A9 91	OPNERR	LDA	#BADMOD	SET ERROR STATUS
F455 B5 4C		STA	DBSTAT	
F457 B5 51	DOPENA	STA	HOLD1	
F459 A5 6A		LDA	RAMTOP	SET UP AN INDIRECT POINTER
I45B B5 65		STA	ADDRESS+1	
I45D BC 45 FE		LDY	ALOCAT,X	ALLOCATE N BLOCKS OF 40 BYTES
I460 A9 2B	DOPENI	LDA	#40	
F462 20 21 F9		JSR	DBSUB	
F465 B8		DEY		
F466 D0 FB		BNE	DOPENI	
I46B AD 6F 02		LDA	OPRIOR	CLEAR OTIA MODES
I46B 29 3F		AND	#3F	
I46D B5 67		STA	OPNTMP+1	
F46F AB		TAY		

I-470 E0 08	CPX	#8	TEST IF 320X1
I-472 90 17	BCC	NOT8	
I-474 8A	TXA		GET 2 LOW BITS
I-475 6A	ROR	A	
I-476 6A	ROR	A	
I-477 6A	ROR	A	
I-478 29 C0	AND	#6C0	NOW 2 TOP BITS
I-47A 05 67	ORA	OPNTMP+1	
I-47C AB	TAY		
I-47D A9 10	LDA	#16	SUBTRACT 16 MORE FOR PAGE BOUNDARY
I-47F 20 21 F9	JSR	DBSUB	
F-482 E0 08	CPX	#11	TEST MODE 11
F-484 D0 05	BNE	NOT8	IF MODE = 11
F-486 A9 06	LDA	#6	PUT OTIA LUM VALUE INTO BACKGROUND REGISTER
F-488 BD C8 02	STA	COLOR4	
F-48B BC 6F 02	STY	OPRIOR	STORE NEW PRIORITY
F-48E A5 64	LDA	ADRESS	SAVE MEMORY SCAN COUNTER ADDRESS
F-490 B5 58	STA	SAVMBC	
F-492 A5 65	LDA	ADRESS+1	
F-494 B5 59	STA	SAVMBC+1	
F-496 AD 08 D4	LDA	VBCOUNT	WAIT FOR NEXT VBLANK BEFORE MESSAGE
F-499 C9 7A	CMF	#7A	WITH THE DISPLAY LIST
F-49B D0 F9	BNE	VBCOUNT	
F-49D 20 1F F9	JSR	DBDEC	START PUTTING DISPLAY LIST RIGHT UNDER RAM
F-4A0 BD 78 FE	LDA	PAGETB, K	TEST IF DISPLAY LIST WILL BE IN TROUBLE
F-4A3 F0 06	BEG	NONOD	OF CROSSING A 256 BYTE PAGE BOUNDARY
F-4A5 A9 FF	LDA	#FF	IF SO, DROP DOWN A PAGE
F-4A7 B5 64	STA	ADRESS	
F-4A9 C6 65	DEC	ADRESS+1	
F-4AB A5 64	LDA	ADRESS	SAVE END OF DISPLAY LIST FOR LATER
F-4AD B5 68	STA	SAVADR	
F-4AF A5 65	LDA	ADRESS+1	
F-4B1 B5 69	STA	SAVADR+1	
F-4B3 20 13 F9	JSR	DBDEC	(DOUBLE BYTE DOUBLE DECREMENT)
F-4B6 A9 41	LDA	#41	(ANTIC) WAIT FOR VBLANK AND JMP TO TOP
F-4BB 20 17 F9	JSR	STORE	
F-4BB B6 66	STX	OPNTMP	
F-4BD A9 18	LDA	#24	INITIALIZE BOTSCR
F-4BF BD BF 02	STA	BOTSCR	
F-4C2 A5 57	LDA	DINDEX	DISALLOW MIXED MODE IF MODE. 0E. 9
F-4C4 C9 09	CMF	#9	
F-4C6 B0 20	BCS	NOTMXD	
F-4C8 A5 2A	LDA	ICAX1Z	TEST MIXED MODE
F-4CA 29 10	AND	#10	
F-4CC F0 27	BEG	NOTMXD	
F-4CE A9 04	LDA	#4	
F-4D0 BD BF 02	STA	BOTSCR	
F-4D3 A2 02	LDY	#2	ADD 4 LINES OF TEXT AT BOTTOM OF SCREEN
F-4D5 A9 02	LDA	#2	
F-4D7 20 17 F9	JSR	STORE	
F-4DA CA	DEX		
F-4DB 10 F8	BPL	DOPEN2	
F-4DD A4 6A	LDY	RANTOP	RELOAD MSC FOR TEXT
F-4DF BB	DEY		
F-4E0 98	TYA		
F-4E1 20 17 F9	JSR	STORE	
F-4E4 A9 60	LDA	#60	
F-4E6 20 17 F9	JSR	STORE	
F-4E9 A9 42	LDA	#42	
F-4EB 20 17 F9	JSR	STORE	

F4EE 18	CLC	
F4EF A9 0C	LDA	#MXDMDE-MUMBLE POINT X AT MIXED MODE TABLE
F4F1 65 66	ADC	OPNTMP
F4F3 85 66	BTA	OPNTMP
F4F5 A1 66	LDY	OPNTMP
F4F7 8E 81 FE	LDX	NUMBLE Y GET NUMBER OF DISPLAY LIST ENTRIES
F4FA A5 91	LDA	HOLD1 STORE N DLE'S
F4FC 20 17 F9	JBR	STORE
F4FF CA	DEX	
F500 D0 FB	BNE	DOPEN3
F502 A5 57	LDA	RINDEX DO THE MESSY 320X1 PROBLEM
F504 C9 08	CMP	#8
F506 90 1C	BCC	DOPEN5
F508 A2 5D	LDX	#93 GET REMAINING NUMBER OF DLE'S
F50A A5 6A	LDA	RAMTOP RELOAD MEMORY SCAN COUNTER
F50C 38	BEC	
F50D E9 10	SBC	#10
F50F 20 17 F9	JBR	STORE
F512 A9 00	LDA	#0
F514 20 17 F9	JBR	STORE
F517 A9 4F	LDA	#4F (ANTIC) RELOAD MSC CODE
F519 20 17 F9	JBR	STORE
F51C A5 51	LDA	HOLD1 DO REMAINING DLE'S
F51E 20 17 F9	JBR	STORE
F521 CA	DEX	
F522 D0 FB	BNE	DOPEN4
F524 A5 59	LDA	SAVMSC+1 POLISH OFF DISPLAY LIST
F526 20 17 F9	JBR	STORE
F529 A5 58	LDA	SAVMSC
F52B 20 17 F9	JBR	STORE
F52E A5 81	LDA	HOLD1
F530 09 40	DRA	#40
F532 20 17 F9	JBR	STORE
F535 A9 70	LDA	#70 24 BLANK LINES
F537 20 17 F9	JBR	STORE
F53A A9 70	LDA	#70
F53C 20 17 F9	JBR	STORE
F53F A5 64	LDA	ADRESS SAVE DISPLAY LIST ADDRESS
F541 8D 30 02	BTA	BDLSTL
F544 A5 65	LDA	ADRESS+1
F546 8D 31 02	BTA	BDLSTL+1
F549 A9 70	LDA	#70 ADD LAST BLANK LINE ENTRY
F54B 20 17 F9	JBR	STORE POSITION ADRESS-BDLSTL+1
F54E A5 64	LDA	ADRESS STORE NEW MEMTOP
F550 8D E5 02	BTA	MEMTOP
F553 A5 65	LDA	ADRESS+1
F555 8D E6 02	BTA	MEMTOP+1
F558 A5 68	LDA	SAVADR
F55A 85 64	BTA	ADRESS
F55C A5 69	LDA	SAVADR+1
F55E 85 65	BTA	ADRESS+1
F560 AD 31 02	LDA	BDLSTL+1
F563 20 17 F9	JBR	STORE
F566 AD 30 02	LDA	BDLSTL
F569 20 17 F9	JBR	STORE
F56C A5 4C	LDA	DBTAT IF ERROR OCCURRED ON ALLOCATION, OPEN THE EDITOR
F56E 10 07	BPL	DOPEN9
F570 48	PHA	SAVE STATUS
F571 20 FC F3	JBR	EOPEN OPEN THE EDITOR
F574 68	PLA	RESTORE STATUS

```

1575 AB          TAY          AND RETURN IT TO CIO
1576 60          RTS
1577 A5 2A      DOPEN9 LDA    ICAX1Z TEST CLEAR INHIBIT BIT
1579 29 20      AND      #620
157D 00 00      BNE     DOPEN7
157D 20 B9 F7   JSR      CLRBCR CLEAR SCREEN
1580 8D 90 02   STA     TXTROW AND HOME TEXT CURSOR (AC IS ZERO)
1583 A5 52      LDA     LMARON
1585 8D 91 02   STA     TXTCOL
1588 A9 22      DOPEN7 LDA     #622 EVERYTHING ELSE IS SET UP
158A 0D 2F 02   ORA     SDMCTL SO TURN ON DMACTL
158D 8D 2F 02   STA     SDMCTL
1590 4C 21 F4   JMP      RETURN
  
```

```

1593 20 96 FA   GETCH JSR     RANGE GETCH DOES INCRBR, GETPLT DOESN'T
1596 20 A2 F5   JSR     GETPLT
1599 20 32 FB   JSR     INATAC CONVERT INTERNAL CODE TO ATASCII
159C 20 D4 F9   JSR     INCRBR
159F 4C 34 F6   JMP     RETURN
15A2 20 47 F9   GETPLT JSR     CONVRT CONVERT ROW/COLUMN TO ADRESS
15A5 B1 64      LDA     (ADRESS),Y
15A7 2D A0 02   AND     DMASK
15AA 44 6F      SHIFTD LSR     SHIFTD SHIFT DATA DOWN TO LOW BITS
15AC 80 03      RCR     SHIFTD
15AE 4A         LSR     A
15AF 10 F9      DPL     SHIFTD (UNCONDITIONAL)
15B1 8D FA 02   SHIFTD STA     CHAR
15B4 C9 00      CMP     #0 RESTORE FLAG8 ALSO
15B6 60         RTS
  
```

```

15B7 8D FB 02   OUTCH BTA     ATACHR
15DA 20 96 FA   JSR     RANGE
15BD AD FB 02   OUTCHA LDA     OFFCRB
15C0 C9 7D      CMP     ATACHR TEST FOR CLEAR SCREEN
15C2 D0 06      BNE     #CLRCOD
15C4 20 B9 F7   JSR     OUTCHE
15C7 4C 21 F6   JMP     CLRBCR
15CA AD FB 02   OUTCHE LDA     ATACHR TEST FOR CARRIAGE RETURN
15CD C9 98      CMP     #CR
15CF D0 06      BNE     OUTCHB
15D1 20 30 FA   JSR     DOCRWB DO CR
15D4 4C 21 F6   JMP     RETURN
15D7 20 E0 F5   OUTCHB JSR     OUTPLT
15DA 20 D8 F9   JSR     INCRBR
15DD 4C 21 F6   JMP     RETURN
  
```

```

15E0 AD FF 02   OUTPLT LDA     SBFLAG *****LOOP HERE IF START/STOP FLAG IS NON-0
15E3 D0 FB      BNE     OUTPLT
15E5 A2 02      LDX     #2
15E7 B5 54      CRLOOP LDA     ROWCRB, X SAVE CURSOR LOCATION FOR DRAW LINE TO DRAW FROM
15E9 95 5A      STA     OLDROW, X
15EB CA         DEX
15EC 10 F9      BPL     CRLOOP
15EE AD FB 02   LDA     ATACHR CONVERT ATASCII(ATACHR) TO INTERNAL(CHAR)
15F1 AB         TAY     SAVE ATACHR
  
```


I-5F2 2A	ROL	A	
I-5F3 2A	ROL	A	
I-5F4 2A	ROL	A	
I-5F5 2A	ROL	A	
I-5F6 29 03	AND	#3	
I-5F8 AA	TAX		X HAS INDEX INTO ATAIN
I-5F9 98	TYA		RESTORE ATACHR
I-5FA 29 9F	AND	#9F	STRIP OFF COLUMN ADDRESS
I-5FC 1D F6 FE	ORA		ATAINT, X OR IN NEW COLUMN ADDRESS
I-5FF 8D FA 02	STA		CHAR
I-602 20 47 F9	JSR		CONVRT
I-605 AD FA 02	LDA		CHAR
I-608 46 6F	LBR		SHFAMT SHIFT UP TO PROPER POSITION
I-60A 80 04	BCS		SHIFT2
I-60C 0A	ASL	A	
I-60D 4C 0B F4	JMP		SHIFTU
I-610 2D A0 02	AND		DMASK
I-613 85 50	STA		TMPCHR SAVE SHIFTED DATA
I-615 AD A0 02	LDA		DMASK INVERT MASK
I-618 47 FF	EOR		#FF
I-61A 31 64	AND		(ADDRESS), Y MASK OFF OLD DATA
I-61C 05 50	ORA		TMPCHR OR IN NEW DATA
I-61E 91 64	STA		(ADDRESS), Y
I-620 60	RTB		
I-621 20 A2 F3	RETURN JSR		GETPLT DO CURSOR ON THE WAY OUT
I-624 85 5D	STA		OLDCHR
I-626 A6 57	LDX		DINDEX GRAPHICS HAVE INVISIBLE CURSOR
I-628 D0 0A	BNE		RETURN
I-62A AE F0 02	LDX		CRSINH TEST CURSOR INHIBIT
I-62D D0 05	BNE		RETURN
I-62F 49 80	EOR		#80 TOGGLE MSB
I-631 20 FF F3	JSR		OUTCH2 DISPLAY IT
I-634 A4 4C	LDY		DSTAT RETURN TO CIO WITH STATUS IN Y
I-636 A9 01	LDA		#SUCCESS
I-638 85 4C	STA		DSTAT SET STATUS= SUCCESSFUL COMPLETION
I-63A AD FB 02	LDA		ATACHR PUT ATACHR IN AC FOR RETURN TO CIO
I-63D 60	NOFUNC RTB		(NON-EXISTENT FUNCTION RETURN POINT)

END OF DISPLAY HANDLER

163E 20 B3 FC	EOETCH	JSR	SWAP	
1641 20 B8 FA		JSR	ERANGE	
1644 A3 6B		LDA	BUFCNT	ANYTHING IN THE BUFFER?
1646 D0 34		BNE	EOETC3	YES
164B A5 54		LDA	ROWCRB	NO, SO SAVE BUFFER START ADDRESS
164A B5 6C		STA	BUFSTR	
164C A5 55		LDA	COLCRB	
164E B5 6D		STA	BUFSTR+1	
1650 20 E2 F6	EOETC1	JSR	EOETCH	LET'S FILL OUR BUFFER
1653 B4 4C		STY	DSTAT	SAVE KEYBOARD STATUS
1655 AD FB 02		LDA	ATACHR	TEST FOR CR
165B C9 9B		CHP	#CR	
165A F0 12		BEQ	EOETC2	
165C 20 AD F6		JSR	DOBS	NO, SO PRINT IT
165F 20 B3 FC		JSR	SWAP	JSR DOBS DID SWAP SO SWAP BACK
1662 A5 63		LDA	LOGCOL	DEEP IF NEARING LOGICAL COL 120
1664 C9 71		CHP	#113	
1666 D0 03		BNE	EOETC6	
166B 20 0A F9		JSR	BELL	
166B 4C 50 F4	EOETC4	JMP	EOETC1	
166E 20 E4 FA	EOETC2	JSR	OFFCRB	GET BUFFER COUNT
1671 20 0D FC		JSR	DOBUFC	
1674 A5 6C		LDA	BUFSTR	RETURN A CHARACTER
1676 B5 54		STA	ROWCRB	
167B A5 6D		LDA	BUFSTR+1	
167A B5 55		STA	COLCRB	
167C A5 6B	EOETC3	LDA	BUFCNT	
167E F0 11		BEQ	EOETC5	
1680 C6 6B	EOETC7	DEC	BUFCNT	AND RETURN TILL BUFCNT=0
1682 F0 0D		BEQ	EOETC5	
1684 A5 4C		LDA	DSTAT	IF ERROR, LOOP ON EOETC7 UNTIL BUFFER IS EMPTIED
1686 30 FB		BHI	EOETC7	
168B 20 93 F5		JSR	EOETCH	
168B 8D FB 02		STA	ATACHR	
168E 4C B3 FC		JMP	SWAP	AND RETURN WITHOUT TURNING CURSOR BACK ON
1691 20 30 FA	EOETC5	JSR	DOCRWB	DO REAL CARRIAGE RETURN
1694 A9 9B		LDA	#CR	AND RETURN EOL
1696 BD FB 02		STA	ATACHR	
1699 20 21 F6		JSR	RETURN	TURN ON CURSOR THEN SWAP
169C B4 4C		STY	DSTAT	SAVE KEYBOARD STATUS
169E 4C B3 FC		JMP	SWAP	AND RETURN THROUGH RETUR1
16A1 6C 64 00	JSRIND	JMP	(ADDRESS)	JSR TO THIS CAUSES JSR INDIRECT
16A4 BD FB 02	EOETCH	STA	ATACHR	SAVE ATACHR VALUE
16A7 20 B3 FC		JSR	SWAP	
16AA 20 B8 FA		JSR	ERANGE	
16AD 20 E4 FA	DOBS	JSR	OFFCRB	TURN OFF CURSOR
16B0 20 BD FC		JSR	TSCTCL	TEST FOR CONTROL CHARACTERS (Z=1 IF CTL)
16B3 F0 09		BEQ	EOETC5	
16B5 0E A2 02	EOETC6	ASL	ESCFLO	ESCFLO ONLY WORKS ONCE
16B8 20 CA F5		JSR	EOETCHE	
16BB 4C B3 FC	ERETN	JMP	SWAP	AND RETURN THROUGH RETUR1
16BE AD FE 02	EOETC5	LDA	DSPFLO	DO DSPFLO AND ESCFLO
16C1 0D A2 02		ORA	ESCFLO	
16C4 D0 EF		BNE	EOETC6	IF NON-0 DISPLAY RATHER THAN EXECUTE IT

```

F6C6 0E A2 02      ASL      EBCFL0
F6C9 E8            INX
F6CA 0D C6 FE      LDA      CNTRLB,X GET DISPLACEMENT INTO ROUTINE
F6CD 05 64          STA      ADDRESS
F6CF 0D C7 FE      LDA      CNTRLB+1,X GET HIGH BYTE
F6D2 05 63          STA      ADDRESS+1
F6D4 20 A1 FA      JBR      JBRIND DO COMPUTED JBR
F6D7 20 21 F6      JBR      RETURN DO CURSOR
F6DA 4C B3 FC      JMP      SWAP ALL DONE SO RETURN THROUGH RETURN1

```

END SCREEN EDITOR

BEGIN KEYBOARD HANDLER

```

F6DD A9 FF      KOETC2 LDA      #0FF
F6DF 0D FC 02      STA      CH
F6E2 A3 2A      KOETCH LDA      ICANIZ TEST LBB OF AUX1 FOR SPECIAL EDITOR READ MODE
F6E4 4A          LBR      A
F6E6 B0 62      BCB      GETONT
F6E7 A9 80      LDA      #BRKABT
F6E9 A6 11      LDX      BRKKEY TEST BREAK
F6EB F0 5B      BEQ      K7 IF BREAK, PUT BRKABT IN DSTAT AND CR IN ATACHR
F6ED AD FC 02      LDA      CH
F6F0 C9 FF      CMC      #0FF
F6F2 F0 EE      BEQ      KOETCH
F6F4 05 7C      STA      HOLDCH SAVE CH FOR SHIFT LOCK PROC
F6F6 A2 FF      LDX      #0FF "CLEAR" CH
F6F8 0E FC 02      STX      CH
F6FB 20 DB FF      JBR      CLICH DO KEYBOARD AUDIO FEEDBACK (A IS OK)
F6FE AA      KOETC3 TAX      DO ABCCON
F6FF E0 C0      CPX      #0C0 TEST FOR CTL & SHIFT TOGETHER
F701 90 02      BCC      ABCC01
F703 A2 03      LDX      #3 BAD CODE
F705 0D FE FE      ASCCO1 LDA      ATABCI,X
F708 0D FB 02      STA      ATACHR DONE
F70B C9 80      CMP      #80 DO NULL
F70D F0 CE      BEQ      KOETC2
F70F C9 81      CMP      #81 CHECK ATARI KEY
F711 00 0B      BNE      KOETC1
F713 AD B6 02      LDA      INVFL0
F716 49 B0      EOR      #80
F718 0D B6 02      STA      INVFL0
F71B 4C DD F6      JMP      KOETC2 DONT RETURN A VALUE
F71E C9 82      KOETC1 CMP      #82 CAPS/LOWER
F720 D0 07      BNE      K1
F722 A9 00      LDA      #0 CLEAR SHFLOK
F724 0D BE 02      STA      SHFLOK
F727 F0 B4      BEQ      KOETC2
F729 C9 83      K1 CMP      #83 SHIFT CAPS/LOWER
F72B D0 07      BNE      K2
F72D A9 40      LDA      #40
F72F 0D DE 02      STA      SHFLOK SHIFT BIT
F732 D0 A9      BNE      KOETC2

```

1734 C9 84	K2	CMP	#B4	CNTL CAPS/LOWER
1736 D0 07		BNE	K3	
1738 A9 80		LDA	#B0	CNTL BIT
173A 8D 0E 02		STA	SHFLOK	
173D D0 7E		BNE	K0ETC2	
173F C9 85	K3	CMP	#B5	DO EOF
1741 D0 0A		BNE	K6	
1743 A9 88		LDA	NEOFERR	
1745 85 4C	K7	STA	DSTAT	
1747 85 11		STA	BRKKEY	RESTORE BREAK
1749 A9 9B	GETOUT	LDA	WCR	PUT CR IN ATACHR
174B D0 26		BNE	K8	(UNCONDITIONAL)
174D A5 7C	K6	LDA	HOLDCH	PROCESS SHIFT LOCKS
174F C9 40		CMP	#40	REGULAR SHIFT AND CONTROL TAKE PRECEDENCE
1751 B0 15		BCS	K5	OVER LOCK
1753 AD FB 02		LDA	ATACHR	TEST FOR ALPHA
1756 C9 61		CMP	#61	LOWER CASE A
1758 90 0E		BCC	K5	NOT ALPHA IF LT
175A C9 78		CMP	#7B	LOWER CASE Z+1
175C B0 0A		BCS	K5	NOT ALPHA IF GE
175E AD BE 02		LDA	SHFLOK	DO SHIFT/CONTROL LOCK
1761 F0 05		BEQ	K5	IF NO LOCK, DONT RE-DO IT
1763 05 7C		ORA	HOLDCH	
1765 4C FE F6		JMP	K0ETC3	DO RETRY
1768 20 BD FC	K9	JBR	TBTCTL	DONT INVERT MSB OF CONTROL CHARACTERS
176B F0 09		BEQ	K4	
176D AD FB 02		LDA	ATACHR	
1770 4D B6 02		EDR	INVFLQ	
1773 BD FB 02	K8	STA	ATACHR	
1776 4C 34 F6	K4	JMP	RETURN	ALL DONE

CONTROL CHARACTER PROCESSORS

1779 A9 80	ESCAPE	LDA	#80	SET ESCAPE FLAG
177B BD A2 02		STA	ESCFL0	
177E 60		RTB		
177F C6 54	CRSRUP	DEC	ROWCRB	
1781 10 06		BPL	COMRET	
1783 AE BF 02		LDX	BOTSCR	WRAPAROUND
1786 CA		DEX		
1787 B6 54	UPDNCH	STX	ROWCRB	
1789 4C 5C FC	COMRET	JMP	STRBEQ	COLVERT ROW AND COL TO LOGCOL AND RETURN
178C E6 54	CRSRDN	INC	ROWCRB	
178E A5 54		LDA	ROWCRB	
1790 CD BF 02		CMP	BOTSCR	
1793 90 F4		BCC	COMRET	
1795 A2 00		LDX	#0	
1797 F0 EE		BEQ	UPDNCH	(UNCONDITIONAL)
1799 C6 55	CRSRLE	DEC	COLCRB	
179B A5 55		LDA	COLCRB	
179D 30 04		BMI	CRSRL1	(IF LMARON=0, THIS ELIMINATED PROBLEM CASE)
179F C5 52		CMP	LMARON	
17A1 B0 04		BCB	COMRE1	
17A3 A5 53	CRSRL1	LDA	RMARON	
17A5 B5 55	LFRTCM	STA	COLCRB	
17A7 4C DD FB	COMRE1	JMP	DOLCOL	COLVERT ROW AND COL TO LOGCOL AND RETURN
17AA E6 55	CRSRRT	INC	COLCRB	
17AC A5 55		LDA	COLCRB	
17AE C5 53		CMP	RMARON	
17B0 90 F5		BCC	COMRE1	
17B2 F0 F3		BEQ	COMRE1	(CAUSE BLE)
17B4 A5 52		LDA	LMARON	
17B6 4C A5 F7		JMP	LFRTCM	UNCONDITIONAL TO COMMON STORE
17B9 20 F3 FC	CLRBCR	JSR	PUTMBC	
17BC A0 00		LDY	#0	
17BE 98		TYA		PUT 0 IN THE AC
17BF 91 64	CLRBC2	STA	(ADREB8),Y (AC IS ZERO)	
17C1 C8		INY		
17C2 D0 F8		BNE	CLRBC2	
17C4 E6 65		INC	ADREB8+1	
17C6 A6 65		LDX	ADREB8+1	
17C8 E4 6A		CPX	RAMTOP	
17CA 90 F3		BCC	CLRBC2	
17CC A9 FF		LDA	#FF	CLEAN UP LOGICAL LINE BIT MAP
17CE 99 B2 02	CLRBC3	STA	LOOMAP,Y (Y IS ZERO AFTER CLRBC2 LOOP)	
17D1 C8		INY		
17D2 C0 04		CPY	#4	
17D4 90 F8		BCC	CLRBC3	
17D6 20 E4 FC	HOME	JSR	COLCR	PLACE COLCRB AT LEFT EDGE
17D9 B5 63		STA	LOGCOL	
17DB B5 6D		STA	BUFSTR+1	
17DD A9 00		LDA	#0	
17DF B5 54		STA	ROWCRB	
17E1 B5 56		STA	COLCRB+1	
17E3 B5 6C		STA	BUFSTR	
17E5 60		RTS		

I7EA F0 21		BEG	BS1	
I7EC A5 55	BSA	LDA	COLCRB	LSFT EDGE?
I7EE C5 52		CMP	LMARGN	
I7FO D0 03		BNE	BS3	NO
I7F2 20 73 FC		JSR	DELTIM	YES, SEE IF LINE SHOULD BE DELETED
I7F5 20 99 F7	BS3	JSR	CRSRLF	
I7FB A5 55		LDA	COLCRB	
I7FA C5 53		CMP	RMARGN	
I7FC D0 07		BNE	BS2	
I7FE A5 54		LDA	ROWCRB	
I800 F0 03		BEG	BS2	
I802 20 7F F7		JSR	CRSRUP	
I805 A9 20	BS2	LDA	#20	MAKE BACKSPACE DESTRUCTIVE
I807 BD FB 02		BTB	ATACHR	
I80A 20 E0 F5		JSR	OUTPLT	
I80D 4C DD FB	BS1	JMP	DOLCOL	AND RETURN
I810 20 AA F7	TAB	JSR	CRSRRT	BEGIN SEARCH
I813 A5 55		LDA	COLCRB	TEST FOR NEW LINE
I815 C5 52		CMP	LMARGN	
I817 D0 0A		BNE	TAB1	NO
I819 20 34 FA		JSR	DOCR	DO CARRIAGE RETURN
I81C 20 20 FB		JSR	LOGGET	CHECK IF END OF LOGICAL LINE
I81F 90 02		BCC	TAB1	NO, CONTINUE
I821 BD 07		BCC	TAB2	(UNCONDITIONAL)
I823 A5 63	TAB1	LDA	LOGCOL	CHECK FOR TAB STOP
I825 20 25 FB		JSR	BITGET	
I828 90 E6		BCC	TAB	NO, SO KEEP LOOKING
I82A 4C DD FB	TAB2	JMP	DOLCOL	COLVERT ROW AND COL TO LOGCOL AND RETURN
I82D A5 63	SETTAB	LDA	LOGCOL	
I82F 4C 06 FB		JMP	BITSET	SET BIT IN MAP AND RETURN
I832 A5 63	CLRTAB	LDA	LOGCOL	
I834 4C 12 FB		JMP	BITCLR	CLEAR
I837 20 9D FC	INSCHR	JSR	PHACRB	
I83A 20 A2 F5		JSR	GETPLT	GET CHARACTER UNDER CURSOR
I83D B5 7D		STA	INSDAT	
I83F A9 00		LDA	NO	
I841 BD BB 02		BTB	SCRFL0	
I844 20 FF FB	INSCH4	JSR	OUTCH2	STORE DATA
I847 A5 63		LDA	LOGCOL	SAVE LOGCOL: IF AFTER INCRSA LOGCOL IS
I849 4B		PHA		< THAN IT IS NOW, END LOOP
I84A 20 DC F9		JSR	INCRSA	SPECIAL INCRSR ENTRY POINT
I84D 6B		PLA		
I84E C5 63		CMP	LOGCOL	
I850 B0 0C		BCC	INSCH3	QUIT
I852 A5 7D	INSCH1	LDA	INSDAT	KEEP GOING
I854 4B		PHA		
I855 20 A2 F5		JSR	GETPLT	
I858 B5 7D		BTB	INSDAT	
I85A 6B		PLA		
I85B 4C 44 FB		JMP	INSCH4	
I85E 20 AB FC	INSCH3	JSR	PLACRB	
I861 CE BB 02	INSCH6	DEC	SCRFL0	
I864 30 04		BMI	INSCH5	IF BSCROLL OCCURRED
I86A C6 54		DEC	ROWCRB	MOVE CURSOR UP
I86B D0 F7		BNE	INSCH6	(UNCOND) CONTINUE UNTIL SCRFL0 IS MINUS
I86A 4C DD FB	INSCH5	JMP	DOLCOL	COLVERT ROW AND COL TO LOGCOL AND RETURN
I86D 20 9D FC		DELCHR	JSR	PHACRB
I870 20 47 F9		DELCHI	JSR	CONVRT GET DATA TO THE RIGHT OF THE CURSOR

1873 A5 64	LDA	ADRESS	
1875 B5 68	STA	SAVADR	SAVE ADDRESS TO KNOW WHERE TO PUT DATA
1877 A5 65	LDA	ADRESS+1	
1879 B5 69	STA	SAVADR+1	
187B A5 63	LDA	LOGCOL	
187D 4B	PHA		
187E 20 04 F9	JSR	INCRB8	PUT CURSOR OVER NEXT CHARACTER
1881 68	PLA		
1882 C5 63	CMP	LOGCOL	TEST NEW LOGCOL AGAINST OLD LOGCOL
1884 B0 10	BCB	DELCH2	IF OLD. GE. NEW THEN QUIT
1886 A5 54	LDA	ROWCR8	IS ROW OFF SCREEN?
1888 C0 BF 02	CMP	ROTBCR	
188B B0 09	BCB	DELCH2	YES, SO QUIT
188D 20 A2 F5	JBR	GETPLT	GET DATA UNDER CURSOR
1890 A0 00	LDY	#0	
1892 91 68	STA	(SAVADR),Y	PUT IT IN PREVIOUS POSITION
1894 F0 DA	BEQ	DELCH1	AND LOOP (UNCONDITIONAL)
1896 A0 00	LDY	#0	
1898 9B	TYA		
1899 91 68	STA	(SAVADR),Y	CLEAR THE LAST POSITION
189B 20 68 FC	JSR	DELTA	TRY TO DELETE A LINE
189E 20 A8 FC	JSR	PLACR8	
18A1 4C DD FB	JMP	DOLCOL	AND RETURN
18A4 3B	INSLIN	SEC	NORMAL INSLIN PUTS "1" INTO BIT MAP
18A5 20 78 FB	INSLIA	JBR	EXTEND ENTRY POINT FOR C=0
18AB A5 52	LDA	LMARON	DO CARRIAGE RETURN (NO LF)
18AA B5 55	STA	COLCR8	
18AC 20 47 F9	JSR	CONVRT	GET ADDRESS
18AF A5 64	LDA	ADRESS	SET UP TO=40+FROM (FROM = CURSOR)
18B1 B5 68	STA	FRMADR	
18B3 1B	CLC		
18B4 69 2B	ADC	#40	
18B6 B5 66	STA	TOADR	
18B8 A5 65	LDA	ADRESS+1	
18BA B5 69	STA	FRMADR+1	
18BC 69 00	ADC	#0	
18BE B5 67	STA	TOADR+1	
18C0 A6 54	LDX	ROWCR8	SET UP LOOP COUNTER
18C2 E0 17	CPX	#23	
18C4 F0 0B	BEQ	INSLI2	
18C6 20 4E FB	INSLI1	JSR	MOVLIN
18C9 EB	INX		
18CA E0 17	CPX	#23	
18CC D0 FB	BNE	INSLI1	
18CE 20 9B FB	INSLI2	JSR	CLRLIN CLEAR CURRENT LINE
18D1 4C DD FB	JMP	DOLCOL	COLVERT ROW AND COL TO LOGCOL AND RETURN
18D4 20 DD FB	JBR	DOLCOL	GET BEGINNING OF LOG LINE (HOLD1)
18D7 A4 51	DELLIA	LDY	HOLD1 SQUEEZE BIT MAP
18D9 B4 54	BTY	ROWCR8	PUT CURSOR THERE
18DB A4 54	DELLIB	LDY	ROWCR8
18DD 9B	DELLI1	TYA	
18DE 3B	SEC		
18DF 20 23 FB	JSR	LO2GET	GET NEXT BIT
18E2 0B	PHP		
18E3 9B	TYA		
18E4 1B	CLC		
18E6 69 7B	ADC	#120	
18E7 2B	PLP		
18E9 20 04 FB	JSR	BITPUT	WRITE IT OVER PRESENT BIT
18EB CB	INY		

DISPLAY HANDLER -- 10-30-78 -- DISPLC

I-BEC C0 18		CPY	#24	
I-BEE D0 ED		BNE	DELL11	LOOP
I-BFO AD B4 02		LDA	LOGMAP+2	SET LBB
I-BF3 09 01		ORA	#1	
I-BF5 BD B4 02		STA	LOGMAP+2	
I-BFB A5 52	DELL12	LDA	LMARGN	DELETE LINE OF DATA USING PART OF SCROLL
I-BFA B5 55		STA	COLCR6	CR NO LF
I-BFC 20 47 F9		JSR	CONVRT	
I-BFF 20 B7 FB		JSR	SCROL1	
I-902 20 20 FB		JSR	LOGOET	TEST NEXT LINE FOR CONTINUATION IS IT A NEW LOG LINE?
I-903 90 D4		BCC	DELL1B	NO SO DELETE ANOTHER
I-907 4C DD FB		JMP	DOLCOL	YES SO DOLCOL AND RETURN
I-90A A0 20	BELL	LDY	#20	
I-90C 20 DB FC	BELL1	JSR	CLICK	
I-90F 88		DEY		
I-910 10 FA		BPL	BELL1	
I-912 60		RTB		

ROUTINES

DOUBLE BYTE DECREMENT OF INDIRECT POINTER
INCLUDING DB SUBTRACT AND DB DOUBLE DECREMENT

I-913 A9 02 DBDDEC LDA #2
I-915 D0 0A BNE DBSUB (UNCONDITIONAL)

STORE DATA INDIRECT AND DECREMENT POINTER
(PLACED HERE TO SAVE JMP DBDDEC AFTER STORE)

I-917 A4 4C STORE LDY DBSTAT RETURN ON ERROR
I-919 30 2B BMI STROK
I-91B A0 00 LDY #0
I-91D 91 64 STORE1 STA (ADDRESS),Y
JMP DBDDEC DECREMENT AND RETURN

I-91F A9 01 DBDDEC LDA #1
I-921 BD 9E 02 DBSUB STA SUBTMP
I-924 A5 4C LDA DBSTAT RETURN ON ERROR
I-926 30 1E BMI STROK

I-928 A5 64 LDA ADREB8
I-92A 38 SEC
I-92B ED 9E 02 SBC SUBTMP
I-92E B5 64 STA ADREB8
I-930 B0 02 BCS DBSUB1
I-932 C6 65 DEC ADREB8+1

I-934 A5 0F DBSUB1 LDA APPMHI+1 MAKE SURE NOTHING EVER OVERWRITES APPMHI
I-936 C5 65 CMP ADREB8+1
I-938 90 0C BCC STROK OK

I-93A D0 06 BNE STRERR ERROR
I-93C A5 0E LDA APPMHI
I-93E C5 64 CMP ADREB8

I-940 90 04 BCC STROK
I-942 A9 93 STRERR LDA #SCREEN SHOW MEN TOO SMALL FOR SCREEN ERROR
I-944 B5 4C STA DBSTAT
I-946 60 STROK RTS

CONVERT ROW/COLUMN CURSOR INTO REAL ADDRESS (FROM SAVMSC ON UP)

I-947 A5 54 CONVRT LDA ROWCRS SAVE CURSOR
I-949 4B PHA
I-94A A5 55 LDA COLCRS

I-94C 4B PHA
I-94D A5 56 LDA COLCRS+1
I-94F 4B PHA

I-950 20 F3 FC JSR PUTMSC
I-953 A5 54 LDA ROWCRS PUT 10*ROWCRS INTO MLTTMP
I-955 B5 66 STA MLTTMP

I-957 A9 00 LDA #0
I-959 B5 67 STA MLTTMP+1
I-95B A5 66 LDA MLTTMP QUICK XB

I-95D 0A ABL A
I-95E 26 67 ROL MLTTMP+1
I-960 B5 51 STA HOLD1 (SAVE 2X VALUE)
I-962 A4 67 LDY MLTTMP+1

1964 8C 9F 02	STY	HOLD2	" "
1967 0A	ASL	A	
1968 26 67	ROL	MLTTMP+1	
196A 0A	ASL	A	
196B 26 67	ROL	MLTTMP+1	
196D 18	CLC	ADD IN 2X	
196E 65 51	ADC	HOLD1	
1970 85 66	STA	MLTTMP	
1972 A5 67	LDA	MLTTMP+1	
1974 6D 9F 02	ADC	HOLD2	
1977 85 67	STA	MLTTMP+1	
1979 A6 87	LDX	DINDEX NOW SHIFT MLTTMP LEFT DINDEX TIMES TO FINISH	
197B 8C 81 FE	LDY	DINDEX, X MULTIPLY	
197E 88	DEY	LOOP N TIMES	
197F 30 07	BMI	CONVR2	
1981 06 66	ASL	MLTTMP	
1983 26 67	ROL	MLTTMP+1	
1985 4C 7E F9	JMP	CONVR1	
198B 8C A5 FE	LDY	DIV2TB, X NOW DIVIDE HCRB TO ACCOUNT FOR PARTIAL BYTES	
198E A5 55	LDA	COLCRB	
198D A2 07	LDX	07 * TRICKY *	
198F 88	DEY		
1990 30 0A	BMI	CONVR4	
1992 CA	DEX		
1993 46 86	LBR	COLCRB+1	
1995 6A	ROR	A	
1996 6E A1 02	ROR	TMPLBT SAVE LOW BITS FOR MASK	
1999 4C 8F F9	JMP	CONVR3	
199C C8	INX	SO Y IS ZERO UPON RETURN FROM THIS ROUTINE	
199D 18	CLC		
199E 65 66	ADC	MLTTMP ADD SHIFTED COLCRB TO MLTTMP	
19A0 85 66	STA	MLTTMP	
19A2 90 02	BCC	CONVR5	
19A4 E6 67	INC	MLTTMP+1	
19A6 3B	SEC	* TRICKY *	
19A7 6E A1 02	CONVR6	TMPLBT SLIDE A BIT UP AGAINST LOW BITS (CONTINUE TILL X--1)	
19AA 18	CLC		
19AB CA	DEX	AND FINISH SHIFT SO LOW BITS ARE	
19AC 10 F9	BPL	CONVR6 RIGHT JUSTIFIED.	
19AE AE A1 02	LDX	TMPLBT IS NOW THE INDEX INTO DMASKTB	
19B1 A5 66	LDA	MLTTMP PREPARE FOR RETURN	
19B3 18	CLC		
19B4 65 66	ADC	ADREB	
19B6 85 66	STA	ADREB	
19B8 85 5E	STA	OLDADR REMEMBER THIS ADDRESS FOR CURSOR	
19BA A5 67	LDA	MLTTMP+1	
19BC 65 65	ADC	ADREB+1	
19BE 85 65	STA	ADREB+1	
19C0 85 8F	STA	OLDADR+1	
19C2 8D 81 FE	LDA	DMASKT, X	
19C5 8D A0 02	STA	DMASK	
19C8 85 6F	STA	SHFAMT	
19CA 6B	PLA		
19CB 85 86	STA	COLCRB+1	
19CD 6B	PLA		
19CE 85 55	STA	COLCRB	
19D0 6B	PLA		
19D1 85 54	STA	ROWCRB	
19D3 60	RTB		

INCREMENT CURSOR AND DETECT BOTH END OF LINE AND END OF SCREEN

19D4 A9 00	INCRB0	LDA	#0	NON-EXTEND ENTRY POINT
19D6 F0 02		BEG	INCRSC	
19DB A9 98	INCRB8	LDA	#98	SPECIAL CASE ELIMINATOR
19DA B5 7D	INCRB8	STA	INSDAT	
19DC E6 63	INCRB8	INC	LOCCOL	(INSCR ENTRY POINT)
19DE E6 55		INC	COLCR8	
19E0 D0 02		RNE	INCRB2	DO HIGH BYTE
19E2 E6 56		INC	COLCR8+1	
19E4 A5 55	INCRB2	LDA	COLCR8	TEST END OF LINE
19E6 A6 57		LDX	DINDEX	
19E8 D0 8D FE		CMF	COLUMN	TEST TABLED VALUE FOR ALL SCREEN MODES
19EB F0 08		BEG	INC2A	DO CR IF EQUAL
19ED E0 00		CPX	#0	MODE 0?
19EF D0 06		RNE	INCRB3	IF NOT, JUST RETURN
19F1 C5 53		CMF	RMARON	TEST AGAINST RMARON
19F3 F0 02		BEG	INCRB3	EQUAL IS OK
19F5 B0 01		RCH	INC2A	IF GREATER THAN, DO CR
19F7 60	INCRB3	RTB		
19FB E0 08	INC2A	CPX	#8	CHECK MODE
19FA 90 04		BCC	DOCR1	NOT 320X1 80 DO IT
19FC A5 56		LDA	COLCR8+1	TEST HSD
19FE F0 F7		BEG	INCRB3	ONLY AT 64 80 DON'T DO IT
1A00 A5 57	DOCR1	LDA	DINDEX	DONT MESS WITH LOGMAP IF NO MODE ZERO
1A02 D0 30		RNE	DOCR	
1A04 A5 63		LDA	LOCCOL	TEST LINE OVERRUN
1A06 C9 51		CMF	#81	
1A08 90 0A		BCC	DOCR1B	IF LESS THAN 81 IT IS DEFINITELY NOT LINE 8
1A0A A5 7D		LDA	INSDAT	
1A0C F0 26		BEG	DOCR	ONLY DO LOG LINE OVERFLOW IF INSDAT < 0
1A0E 20 30 FA		JSR	DOCRWS	LOG LINE OVERFLOW IS SPECIAL CASE
1A11 4C 77 FA		JMP	INCRB1	RETURN
1A14 20 34 FA	DOCR1B	JSR	DOCR	GET IT OVER WITH
1A17 A5 64		LDA	ROWCR8	
1A19 18		CLC		TEST LOGICAL LINE BIT MAP
1A1A 69 7B		ADC	#120	
1A1C 20 25 FB		JSR	BITOET	
1A1F 90 08		BCC	DOCR1A	DONT EXTEND IF OVERRUN IS INTO MIDDLE OF LOG LINE
1A21 A5 7D		LDA	INSDAT	DONT EXTEND IF INSDAT IS ZERO
1A23 F0 04		BEG	DOCR1A	(INSCR SPECIAL CASE)
1A25 18		CLC		INSERT "0" INTO BIT MAP
1A26 20 A5 FB		JSR	INBL1A	
1A29 4C DD FB	DOCR1A	JMP	DOLCOL	CONVERT ROW AND COL TO LOCCOL AND RETURN
1A2C A9 00	NOSCR1	LDA	#0	DOCR WITHOUT SCROLL
1A2E F0 02		BEG	NOSCR1	(UNCONDITIONAL)
1A30 A9 98	DOCRWS	LDA	#98	DOCR WITH SCROLLING (NORMAL MODE)
1A32 B5 7D	NOSCR1	STA	INSDAT	
1A34 20 E4 FC	DOCR	JSR	COLCR	PLACE COLCR8 AT LEFT EDGE
1A37 A9 00		LDA	#0	
1A39 B5 56		STA	COLCR8+1	
1A3B E6 56		INC	ROWCR8	
1A3D A6 57	DOCR2	LDX	DINDEX	
1A3F A0 18		LDY	#24	SET UP SCROLL LOOP COUNTER
1A41 24 78		BIT	SHPFLO	
1A43 10 05		BPL	DOCR2A	BRANCH IF NORMAL
1A45 A0 04		LDY	#4	
1A47 98		TYA		
1A48 D0 03		BNE	DOCR2B	(UNCONDITIONAL)

```

I-A4A BD 99 FE      DOCR2A LDA    NOROWS, X GET NO OF ROWS
I-A4D C5 54        DOCR2B CMP    ROWCRB
I-A4F D0 26         BNE    INCRB1
I-A51 BC 9D 02      STY    HOLD3
I-A54 BA           TXA           DONT SCROLL IF MODE <> 0
I-A55 D0 20         BNE    INCRB1
I-A57 A5 7D         LDA    INSDAT OR IF INSDAT = 0
I-A59 F0 1C         BEQ    INCRB1
                     LDA    INSDAT IF INSDAT <> #9B THEN ROLL IN A 0
                     CMP    #9B   TO EXTEND BOTTOM LOGICAL LINE
I-A5B C9 9B         SEC
I-A5D 3B           BEQ    DOCR4B
I-A5E F0 01         CLC
I-A60 1B           JSR    DOCR4B
I-A61 20 AC FB      DOCR4B SCROLL LOOP BACK TO HERE IF >1 SCROLLS
I-A64 EE BB 02      INC    BCRFLQ
I-A67 C6 AC         DEC    BUFSTR ROWS MOVE UP SO BUFSTR SHOULD TOO
I-A69 CE 9D 02      DEC    HOLD3
I-A6C AD B2 02      LDA    LOGMAP
I-A6F 3B           SEC           FOR PARTIAL LINES, ROLL IN A "1"
I-A70 10 EF         BPL    DOCR4B AGAIN IF PARTIAL LOGICAL LINE
I-A72 AD 9D 02      LDA    HOLD3
I-A75 B5 54         STA    ROWCRB
I-A77 4C DD FB      INCRB1 JMP    DOLCOL COLVERT ROW AND COL TO LOGCOL AND RETURN

```

SUBEND: SUBTRACT ENDP1 FROM ROWAC OR COLAC: (X=0 OR 2)

```

I-A7A 3B           SUBEND SEC
I-A7B B5 70        LDA    ROWAC, X
I-A7D E5 74        SBC    ENDP1
I-A7F 95 70        STA    ROWAC, X
I-A81 B5 71        LDA    ROWAC+1, X
I-A83 E5 75        SBC    ENDP1+1
I-A85 95 71        STA    ROWAC+1, X
I-A87 60           RTS

```

RANGE: DO CURSOR RANGE TEST. IF ERROR, POP STACK TWICE AND JMP RETURN.
(RANGE IS EDITOR ENTRY POINT AND TEST IF EDITOR IS OPEN.
IF IT ISNT IT OPENS THE EDITOR AND CONTINUES)

```

I-A8B AD BF 02      RANGE LDA    BOTSCR IN BOTSCR=4
I-A8B C9 04        CMP    #4
I-A8D F0 07        BEQ    RANGE THEN IT IS IN MIXED MODE AND OK
I-A8F A5 57        LDA    DINDEX IF MODE = 0
I-A91 F0 03        BEQ    RANGE THEN IT IS IN EDITOR MODE AND OK
I-A93 20 FC F3      JSR    EOPEN IF NOT, OPEN EDITOR
I-A96 A9 27        RANGE LDA    #39 ***** RANGE CHECK RMARON ***** SET UP A0
I-A98 C5 53        CMP    RMARON ***** RANGE CHECK RMARON ***** COMPARE
I-A9A B0 02        BCB    RANGE3 ***** RANGE CHECK RMARON ***** BRANCH OE
I-A9C B5 53        STA    RMARON ***** RANGE CHECK RMARON ***** BAD SO STORE 39
I-A9E A6 57        RANGE3 LDX    DINDEX
I-AA0 BD 99 FE      LDA    NOROWS, X CHECK ROWS
I-AA3 C5 54        CMP    ROWCRB
I-AA5 90 2A        BCC    RNCERR (ERROR IF TABLE GE ROWCRB)
I-AA7 F0 28        BEQ    RNCERR
I-AA9 B0 08        CPX    #0 CHECK FOR 320X1
I-AAB D0 0A        BNE    RANGE1 SPECIAL CASE IT
I-AAD A5 56        LDA    COLCRB+1
I-AAF F0 13        BEQ    RNOOK IF HIGH BYTE IS 0, COL IS OK

```

```

IAB1 C9 01      CMP      #1
IAB3 D0 1C      BNE      RNOERR IF >1, BAD
IAB5 F0 04      BEQ      RNOERR IF 1, GO CHECK LOW BYTE
IAB7 A5 56      RANGE1 LDA      COLC1 FOR OTHERS, NON-ZERO HIGH BYTE IS BAD
IAB9 D0 16      BNE      RNOERR
IAB8 BD BD FE    RANGE2 LDA      COLUMN, X CHECK LOW BYTE
IABE C5 05      CMP      COLCRB
IAC0 90 0F      JCC      RNOERR
IAC2 F0 0D      BEQ      RNOERR
IAC4 A9 01      RNOOK  LDA      #SUCCEB SET STATUS OK
IAC6 B5 4C      STA      DSTAT
IAC8 A9 80      LDA      #BRKABT PREPARE BREAK ABORT STATUS
IACA A6 11      LDX      BRKKEY CHECK BREAK KEY FLAG
IACC B5 11      STA      BRKKEY 'CLEAR' BREAK
IACE F0 06      BEQ      RNOER2 IF BREAK, QUIT IMMEDIATELY AND RETURN TO CIO
IAD0 60      RTS
IAD1 20 D6 F7    RNOERR JSR      HOME ON RANGE ERROR, BRING CURSOR BACK
IAD4 A7 8D      LDA      #CRBROR SHOW CURSOR OVERRANGE ERROR
IAD6 B5 4C      RNOER2 STA      DSTAT
IAD8 68      RNOER1 PLA      RESTORE STACK (THIS ROUTINE IS ALWAYS 1 LEVEL
IAD9 68      PLA      AWAY FROM RETURN TO CIO)
IADA A5 78      LDA      SWPFL0 IF SWAPPED, SWAP BACK
IADC 10 03      BPL      RETUR3
IADE 20 B9 FC    JSR      SWAPA AND DONT DO RETUR1
IAE1 4C 34 F6    RETUR3 JMP      RETUR1 RETURN TO CIO

```

OFFCRS: RESTORE OLD DATA UNDER CURSOR SO IT CAN BE MOVED

```

IAE4 A0 00      OFFCRB LDY      #0
IAE6 A5 3D      LDA      OLDCHR
IAE8 91 5E      STA      (OLDADR),Y
IAEA 60      RTS

```

BITHAP ROUTINES:

```

BITCON: PUT MASK IN BITMSK AND INDEX IN X
BITPUT: PUT CARRY INTO BITHAP
BITROL: ROL CARRY INTO BOTTOM OF BITHAP (SCROLL)
BITSET: SET PROPER BIT
BITCLR: CLEAR PROPER BIT
BITGET: RETURN CARRY SET IF BIT IS THERE
LOGGET: DO BITGET FOR LOGMAP INSTEAD OF TABMAP

```

```

IAEB A8      BITCON PHA
IAEC 29 07    AND      #7
IAEE AA      TAX      GET MASK
IAEF 8D B9 FE LDA      MASKTB,X
IAF2 B5 6E    STA      BITMSK
IAF4 68      PLA      PROCESS INDEX
IAF5 4A      LSR      A
IAF6 4A      LSR      A
IAF7 4A      LSR      A
IAFB AA      TAX
IAF9 60      RTS

```

I-AFA 2E B4 02	BITROL	ROL	LOGMAP+2
I-AFD 2E B3 02		ROL	LOGMAP+1
I-B00 2E B2 02		ROL	LOGMAP
I-B03 60		RTS	

I-B04 90 0C	BITPUT	BCC	BITCLR AND RETURN OTHERWISE FALL THROUGH TO BITSET AND RETURN
-------------	--------	-----	--

I-B06 20 EB FA	BITSET	JSR	BITCON
I-B09 BD A3 02		LDA	TABMAP, X
I-B0C 05 6E		ORA	BITMSK
I-B0E 9D A3 02		STA	TABMAP, X
I-B11 60		RTS	

I-B12 20 EB FA	BITCLR	JSR	BITCON
I-B15 A3 6E		LDA	BITMSK
I-B17 49 FF		EOR	#FF
I-B19 3D A3 02		AND	TABMAP, X
I-B1C 9D A3 02		STA	TABMAP, X
I-B1F 60		RTS	

I-B20 A5 54	LOGGET	LDA	ROWCRB
I-B22 18	LOIGET	CLC	

I-B23 69 78	LOGGET	ADC	#120
I-B25 20 EB FA	BITGET	JSR	BITCON
I-B28 18		CLC	

I-B29 BD A3 02		LDA	TABMAP, X
I-B2C 25 6E		AND	BITMSK
I-B2E F0 01		BEG	BITGET

I-B30 38		BEC	
I-B31 60	BITGET	RTS	

INATAC: INTERNAL(CHAR) TO ATAC(CHAR) CONVERSION

I-B32 AD FA 02	INATAC	LDA	CHAR
I-B35 A4 57		LDV	DINDEX IF GRAPHICS MODES
I-B37 C0 03		CPY	#3
I-B39 B0 0F		BCS	INATA1 THEN DONT CHANGE CHAR

I-B3B 2A		ROL	A
I-B3C 2A		ROL	A
I-B3D 2A		ROL	A
I-B3E 2A		ROL	A
I-B3F 29 03		AND	#3
I-B41 AA		TAX	

I-B42 AD FA 02		LDA	CHAR
I-B45 29 9F		AND	#9F
I-B47 1D FA FE		ORA	INATA, X
I-B4A BD FB 02	INATA1	STA	ATACHR
I-B4D 60		RTS	

MOVLIN: MOVE 40 BYTES AT FRMADR TO TOADR SAVING OLD TOADR
DATA IN THE LINBUF. THEN MAKE NEXT FRMADR
BE AT LINBUF FOR NEXT TRANSFER & TOADR=TOADR+40

I-B4E A9 02	MOVLIN	LDA	#LINBUF/256	SET UP ADDRESS=LINBUF-#247
-------------	--------	-----	-------------	----------------------------


```

1B50 B5 65      STA      ADDRESS+1
1B52 A9 47      LDA      #LINDUF
1B54 B5 64      STA      ADDRESS
1B56 A0 27      LDY      #39
1B58 B1 65      MOVL11   LDA      (TOADR),Y      SAVE TO DATA
1B5A B5 50      STA      TMPCHR
1B5C B1 68      LDA      (FRMADR),Y      STORE DATA
1B5E 91 66      STA      (TOADR),Y
1B60 A5 50      LDA      TMPCHR
1B62 91 64      STA      (ADDRESS),Y
1B64 BB         DEY
1B66 10 F1      BPL      MOVL11
1B68 A5 65      LDA      ADDRESS+1      SET UP FRMADR=LAST LINE
1B6A B5 67      STA      FRMADR+1
1B6C A5 64      LDA      ADDRESS
1B6E B5 68      STA      FRMADR
1B70 1B         CLC      ADD 40 TO TOADR
1B72 A5 66      LDA      TOADR
1B74 69 28      ADC      #40
1B76 B5 66      STA      TOADR
1B78 90 02      BCC      MOVL12
1B7A E6 67      INC      TOADR+1
1B7C 60         MOVL12   RTS

```

/ EXTEND: EXTEND BIT MAP FROM ROWCRB (EXTEND LOGICAL LINE

```

1B7B 0B         EXTEND   PHP      SAVE CARRY
1B7C A0 17      LDY      #23
1B7E 98         EXTEN1   TYA
1B80 20 22 FB   JBR      1010ET
1B82 0B         PIP
1B84 98         TYA
1B86 1B         CLC
1B88 69 79      ADC      #121
1B8A 28         PLP
1B8C 20 04 FB   JBR      BITPUT
1B8E 0B         EXTEN3   DEY
1B90 30 04      BMI      EXTEN4
1B92 C4 54      CPY      ROWCRB
1B94 B0 EC      BCB      EXTEN1
1B96 A5 54      EXTEN4   LDA      ROWCRB
1B98 1B         CLC
1B9A 69 78      ADC      #120
1B9C 28         PLP
1B9E 4C 04 FB   JMP      BITPUT  STORE NEW LINE'S BIT AND RETURN

```

/ CLRLIN: CLEAR LINE CURSOR IS ON

```

1B9B A5 52      CLRLIN   LDA      LMARON
1B9D B5 55      STA      COLCRB
1B9F 20 47 F9   JBR      CONVRT
1BA3 A0 27      LDY      #39
1BA5 A9 00      LDA      #0
1BA7 91 64      CLRL11   STA      (ADDRESS),Y
1BA9 BB         DEY
1BAB 10 FB      DPL      CLRL11

```

I-BAB 60

RTS

SCROLL: SCROLL SCREEN

I-BAC 20 FA FA	SCROLL	JBR	BITROL	ROLL IN CARRY
I-BAF A5 58		LDA	SAVMS	SET UP WORKING REGISTERS
I-BB1 B5 64		STA	ADDRESS	
I-BB3 A5 59		LDA	SAVMS+1	
I-BB5 B5 65		STA	ADDRESS+1	
I-BB7 A0 28	SCROL1	LDY	#40	LOOP
I-BB9 B1 64		LDA	(ADDRESS),Y	
I-BBB A6 6A		LDX	RANTOP	TEST FOR LAST LINE
I-BBD CA		DEX		
I-BBE E4 65		CPX	ADDRESS+1	
I-BC0 D0 08		BNE	SCROL2	
I-BC2 A2 D7		LDX	#D7	
I-BC4 E4 64		CPX	ADDRESS	
I-BC6 B0 02		BCB	SCROL2	
I-BC8 A9 00		LDA	#0	YES SO STORE ZERO DATA FOR THIS ENTIRE LINE
I-BCA A0 00	SCROL2	LDY	#0	
I-BCC 91 64		STA	(ADDRESS),Y	
I-BCE E6 64		INC	ADDRESS	
I-BD0 D0 E5		BNE	SCROL1	
I-BD2 E6 65		INC	ADDRESS+1	
I-BD4 A5 65		LDA	ADDRESS+1	
I-BD6 C5 6A		CHP	RANTOP	
I-BD8 D0 DD		BNE	SCROL1	
I-BDA AC DD FB		JMP	DOLCOL	AND RETURN

DOLCOL: DO LOGICAL COLUMN FROM BITMAP AND COLCRB

I-BDB A9 00	DOLCOL	LDA	#0	START WITH ZERO
I-BDF B5 63		STA	LOGCOL	
I-BE1 A5 54		LDA	ROWCRB	
I-BE3 B5 51		STA	HOLD1	
I-BE5 A5 51	DOLCO1	LDA	HOLD1	ADD IN ROW COMPONENT
I-BE7 20 22 FB		JBR	LOISET	
I-BEA B0 00		BCB	DOLCO2	FOUND BEGINNING OF LINE
I-BEC A5 63		LDA	LOGCOL	ADD 40 AND LOOK BACK ONE
I-BEE 18		CLC		
I-BEF 69 28		ADC	#40	
I-BF1 B5 63		STA	LOGCOL	
I-BF3 C6 51		DEC	HOLD1	UP ONE LINE
I-BF5 4C E5 FB		JMP	DOLCO1	
I-BF7 18	DOLCO2	CLC		ADD IN COLCRB
I-BF9 A5 63		LDA	LOGCOL	
I-BFB 65 55		ADC	COLCRB	
I-BFD B5 63		STA	LOGCOL	
I-BFF 60		RTS		

DOBUFC: COMPUTE BUFFER COUNT AS THE NUMBER OF BYTES FROM
BUFBTR TO END OF LOGICAL LINE WITH TRAILING SPACES REMOVED

I-C00 20 9D FC DOBUFC JSR PHACRB

```

FC03 A5 63      LDA      LOGCOL
FC03 48          PHA
FC06 A5 6C      LDA      BUFSTR START
FC08 B5 54      STA      ROWCRS
FC0A A5 6D      LDA      BUFSTR+1
FC0C B5 55      STA      COLCRS
FC0E A9 01      LDA      #1
FC10 B5 68      STA      BUFCNT
FC12 A2 17      DOBUF1  LDX      #23      NORMAL
FC14 A5 78      LDA      SWPFL0 IF SWAPPED, ROW 3 IS THE LAST LINE ON SCREEN
FC16 10 02      BPL      DOB1
FC18 A2 03      LDX      #3
FC1A E4 54      DOB1     CPX      ROWCRS TEST IF CURR IS AT LAST SCREEN POSITION
FC1C D0 08      BNE      DOBU1A
FC1E A5 55      LDA      COLCRS
FC20 C5 53      CMP      RMARON
FC22 D0 05      BNE      DOBU1A
FC24 E6 6B      INC      BUFCNT YES SO FARE INCRSR TO AVOID SCROLLING
FC26 4C 39 FC    JMP      DOBUF2
FC28 20 D4 F9    DOBU1A  JSR      INCRSR
FC2C E6 6B      INC      BUFCNT
FC2E A5 63      LDA      LOGCOL
FC30 C5 52      CMP      LMARON
FC32 D0 06      BNE      DOBUF1 NOT YET EOL
FC34 C4 54      DEC      ROWCRS BACK UP ONE INCRSR
FC36 20 99 F7    JSR      CRGRLF
FC38 20 A2 F5    DOBUF2  JSR      GETPLT TEST CURRENT COLUMN FOR NON-ZERO DATA
FC3C D0 17      BNE      DOBUF4 QUIT IF NON-ZERO
FC3E C6 68      DEC      BUFCNT DECREMENT COUNTER
FC40 A5 63      LDA      LOGCOL BEGINNING OF LOGICAL LINE YET?
FC42 C5 52      CMP      LMARON
FC44 F0 0F      BEQ      DOBUF4 YES, SO QUIT
FC46 20 99 F7    JSR      CRGRLF BACK UP CURSOR
FC48 A5 55      LDA      COLCRS IF LOGCOL=RMARON, GO UP 1 ROW
FC4B C5 53      CMP      RMARON
FC4D D0 02      BNE      DOBUF3
FC4F C6 64      DEC      ROWCRS
FC51 A5 68      DOBUF3  LDA      BUFCNT
FC53 D0 E4      BNE      DOBUF2 LOOP UNLESS BUFCNT JUST WENT TO ZERO
FC55 68          DOBUF4  PLA
FC56 B5 63      STA      LOGCOL
FC58 20 AB FC    JSR      PLACRS
FC5B 60          RTS

```

STRREQ: MOVE BUFSTR TO BEGINNING OF LOGICAL LINE

```

FC59 20 DD FB    STRREQ  JSR      DOLCOL USE DOLCOL TO POINT HOLD1 AT BOL
FC5F A5 51      LDA      HOLD1
FC61 B5 6C      STA      BUFSTR
FC63 A5 52      LDA      LMARON
FC65 B5 6D      STA      BUFSTR+1
FC67 60          RTS

```

DELTIM: TIME TO DELETE A LINE IF IT IS EMPTY AND AN EXTENSION

IC6B A5 63	DELTIA	LDA	LOGCOL	IF LOGCOL < LMARON
IC6A C5 52		CMP	LMARON	THEN DONT MOVE UP ONE
IC6C D0 02		BNE	DELTID	LINE BEFORE TESTING DELTIM
IC6E C6 54		DEC	ROWCRB	
IC70 20 D0 FB	DELTIB	JSR	DOLCOL	
IC73 A5 63	DELTIM	LDA	LOGCOL	TEST FOR EXTENSION
IC75 C5 52		CMP	LMARON	
IC77 F0 13		REQ	DELTID	NO
IC79 20 47 F9		JSR	CONVRT	
IC7C A5 53		LDA	RMARON	SET UP COUNT
IC7E 38		BEC		
IC7F E3 52		BBC	LMARON	
IC81 A8		TAY		
IC82 D1 64	DELTII	LDA	(ADRE88),Y	
IC84 D0 06		DNE	DELTID	FOUND A NON-0 SO QUIT AND RETURN
IC85 8B		DEY		
IC87 10 F9		BPL	DELTII	
IC89 4C DB FB	DELTID	JMP	DELTIB	DELETE A LINE AND RETURN
IC8C 60	DELTID	RTS		

TSTCTL: SEARCH CNTRLB TABLE TO SEE IF ATACHR IS A CNTRL CHAR

IC8D A2 2D	TSTCTL	LDX	#45	PREPARE TO SEARCH TABLE
IC8F BD C6 FE	TSTCT1	LDA	CNTRLB,X	
IC92 CD FB 02		CMP	ATACHR	
IC95 F0 05		BEG	TSTCT2	
IC97 CA		DEX		
IC98 CA		DEX		
IC99 CA		DEX		
IC9A 10 F3		BPL	TSTCT1	
IC9C 60	TSTCT2	RTB		

PUSH ROWCRB, COLCRB AND COLCRB+1

IC9D A2 02	PHACRB	LDX	#2
IC9F 85 54	PHACR1	LDA	ROWCRB,X
ICA1 9D 8B 02		STA	TMPROW,X
ICA4 CA		DEX	
ICA5 10 FB		BPL	PHACR1
ICA7 60		RTB	

PULL COLCRB+1, COLCRB AND ROWCRB

ICAB A2 02	PLACRB	LDX	#2
ICAA BD 8B 02	PLACR1	LDA	TMPROW,X
ICAD 95 54		STA	ROWCRB,X
ICAF CA		DEX	
ICB0 10 FB		BPL	PLACR1
ICB2 60		RTS	

SWAP: IF MIXED MODE, SWAP TEXT CURSORB WITH REGULAR CURSORB

ICB3 20 B9 FC	SWAP	JBR	SWAPA	THIS ENTRY POINT DOES RETURN
ICB6 4C 34 F6		JMP	RETURN	
ICD9 AD BF 02	SWAPA	LDA	BOTSCR	
ICDC C9 1B		CMP	#24	
ICDE F0 17		BEQ	SWAP3	
ICC0 A2 00		LDX	#11	
ICC2 B5 54	SWAP1	LDA	ROWCRS1	
ICC4 4B		PHA		
ICC5 0D 90 02		LDA	TXTRW, X	
ICCB 95 54		STA	ROWCRS, X	
ICCA 6B		PLA		
ICCB 9D 90 02		STA	TXTRW, X	
ICCE CA		DEX		
ICCF 10 F1		RPL	SWAP1	
ICD1 A5 7B		LDA	SWPFLO	
ICD3 47 FF		EOR	#FF	
ICD5 B5 7B		STA	SWPFLO	
ICD7 60	SWAP3	RTB		

CLICK: MAKE CLICK THROUGH KEYBOARD SPEAKER

ICDB A2 7F	CLICK	LDX	#7F
ICDA 0E 1F D0	CLICK1	STX	CONSOLE
ICDD 0E 0A D4		STX	HSYNC
ICE0 CA		DEX	
ICE1 10 F7		BPL	CLICK1
ICE3 60		RTB	

COLCR: PUTS EITHER 0 OR LMARON INTO COLCRS BASED ON MODE AND SWPFLO

ICE4 A9 00	COLCR	LDA	#0
ICE6 A6 7B		LDX	SWPFLO
FCEB D0 04		BNE	COLCR1
FCEA A4 57		LDX	DINDEX
FCEC D0 02		BNE	COLCR2
ICEE A5 52	COLCR1	LDA	LMARON
ICF0 B5 55	COLCR2	STA	COLCRS
ICF2 60		RTB	

PUTMBC: PUT SAVMBC INTO ADDRESS

ICF3 A5 5B	PUTMBC	LDA	SAVMBC	SET UP ADDRESS
ICF5 B5 64		STA	ADDRESS	
ICF7 A5 59		LDA	SAVMBC+1	
ICF9 B5 65		STA	ADDRESS+1	
ICF8 60		RTB		

```

DRAW -- DRAW A LINE FROM OLDROW,OLDCOL TO NEWROW,NEWCOL
( THE AL MILLER METHOD FROM BASKETBALL )

I-CFC A2 00      DRAW      LDX      #0
I-CFE A5 22      LDA      ICCOMZ  TEST COMMAND: #11-DRAW  #12-FILL
I-D00 C9 11      CMP      #11
I-D02 F0 08      BEQ      DRAWA
I-D04 C9 12      CMP      #12  TEST FILL
I-D06 F0 03      BEQ      DRAWB  YES
I-D08 A0 84      LDY      #INVALID NO, SO RETURN INVALID COMMAND
I-D0A 60          RTS
I-D0B E8          DRAWB  INX
I-D0C 8E 87 02   DRAWA  STX      FILELO
I-D0F A5 54      LDA      ROWCR8  PUT CURSOR INTO NEWROW,NEWCOL
I-D11 85 60      STA      NEWROW
I-D13 A5 55      LDA      COLCR8
I-D15 85 61      STA      NEWCOL
I-D17 A5 66      LDA      COLCR8+1
I-D19 85 62      STA      NEWCOL+1
I-D1B A9 01      LDA      #1
I-D1D 85 79      STA      ROWINC  SET UP INITIAL DIRECTIONS
I-D1F 85 7A      STA      COLINC

F-D21 38          SEC
F-D22 A5 60      LDA      NEWROW  DETERMINE DELTA ROW
F-D24 E9 8A      SBC      OLDROW
I-D26 85 76      STA      DELTAR
I-D28 80 00      BCS      DRAW1  DO DIRECTION AND ABSOLUTE VALUE
I-D2A A7 FF      LDA      #FF      BORROW HAS ATTEMPTED
F-D2C 85 79      STA      ROWINC  SET DIRECTION=DOWN
I-D2E A5 76      LDA      DELTAR
F-D30 A7 FF      EOR      #FF      DELTAR = |DELTAR|
F-D32 18          CLC
I-D33 69 01      ADC      #1
I-D35 85 76      STA      DELTAR

F-D37 38          DRAW1  SEC
F-D38 A5 61      LDA      NEWCOL  NOW DELTA COLUMN
F-D3A E5 58      SBC      OLDCOL
I-D3C 85 77      STA      DELTAC
I-D3E A5 62      LDA      NEWCOL+1  TWO-BYTE QUANTITY
I-D40 E5 5C      SBC      OLDCOL+1
F-D42 85 78      STA      DELTAC+1
I-D44 80 16      BCS      DRAW2  DIRECTION AND ABSOLUTE VALUE
I-D46 A7 FF      LDA      #FF      BORROW HAS ATTEMPTED
I-D48 85 7A      STA      COLINC  SET DIRECTION = LEFT
I-D4A A5 77      LDA      DELTAC
I-D4C A7 FF      EOR      #FF      DELTAC = |DELTAC|
F-D4E 85 77      STA      DELTAC
F-D50 A5 78      LDA      DELTAC+1
F-D52 A7 FF      EOR      #FF
F-D54 85 78      STA      DELTAC+1
F-D56 E6 77      INC      DELTAC  ADD ONE FOR TWO'S COMPLEMENT
F-D58 D0 02      BNE      DRAW2
F-D5A E6 78      INC      DELTAC+1
F-D5C A2 02      DRAW2  LDX      #2  ZERO RAM FOR DRAW LOOP
F-D5E A0 00      LDY      #0
I-D60 84 73      STY      COLAC+1
F-D62 98          DRAW3A TYA
F-D63 95 70      STA      ROWAC, X
F-D65 85 5A      LDA      OLDROW, X

```

I-D67 98 54	STA	ROWCRB, X	
I-D69 CA	DEY		
I-D6A 10 F6	STA	DRAW3A	IF LARGER ONE (ROW OR COL)
I-D6C A5 77	LDA	DELTAC	COMPARE COUNTR AND ENDP
	STA	COUNTR	
	STA	ENDPT	
I-D6E E8	INX		
I-D6F AB	TAY		
I-D70 A5 7B	LDA	DELTAC	
I-D72 B5 7F	STA	COUNTR+1	
I-D74 05 75	STA	ENDPT+1	
I-D76 D0 08	BNE	DRAW3	AUTOMATICALLY LARGER IF MBD>0
I-D78 A5 77	LDA	DELTAC	
I-D7A C5 7A	CHP	DELTAR	LOW COL > LOW ROW?
I-D7C B0 05	BCC	DRAW3	YES
I-D7E A5 76	LDA	DELTAR	
I-D80 02 02	LDX	#2	
I-D82 AB	TAY		
I-D83 98	DRAW3	TVA	PUT IN INITIAL CONDITIONS
I-D84 05 7E	STA	COUNTR	
I-D86 17 74	STA	ENDPT	
I-D88 4B	PHA		SAVE AC
I-D89 A5 75	LDA	ENDPT+1	PUT LBB OF HIGH BYTE
I-D8B 4A	LBR	A	INTO CARRY
I-D8C 6B	PLA		RESTORE AC
I-D8D 6A	ROR	A	ROR THE 9 BIT ACCUMULATOR
I-D8E 95 70	STA	ROWAC, X	
I-D90 A5 7E	DRAW4A	LDA	COUNTR
I-D92 05 7F	ORA	COUNTR+1	TEST ZERO
I-D94 D0 03	BNE	DRAW11	IF COUNTR IS ZERO, LEAVE DRAW
I-D96 4C 42 FE	JMP	DRAW10	
I-D97 1B	DRAW11	CLC	ADD ROW TO ROWAC (PLOT LOOP)
I-D9A A5 70	LDA	ROWAC	
I-D9C 65 76	ADC	DELTAR	
I-D9E B5 70	STA	ROWAC	
I-DA0 90 02	BCC	DRAW5	
I-DA2 E6 71	INC	ROWAC+1	
I-DA4 A5 71	DRAW5	LDA	ROWAC+1
I-DA6 C5 73	CHP	ENDPT+1	COMPARE ROW TO ENDP
I-DAB 90 14	BCC	DRAW6	IF HIGH BYTE OF ROW IS LT. HIGH
I-DA8 D0 06	BNE	DRAW3A	BYTE OF ENDP, BLT TO COLUMN
I-DAC A5 70	LDA	ROWAC	
I-DAE C5 74	CHP	ENDPT	LOW BYTE
I-DB0 90 0C	BCC	DRAW6	ALSO BLT
I-DB2 1B	DRAW5A	CLC	OE SO MOVE POINT
I-DB3 A5 54	LDA	ROWCRB	
I-DB5 65 79	ADC	ROWINC	
I-DB7 05 54	STA	ROWCRB	
I-DB9 A2 00	LDX	#0	AND SUBTRACT ENDP FROM ROWAC
I-DBB 20 7A FA	JSR	SUBEND	
I-DBE 1B	DRAW6	CLC	DO SAME FOR COLUMN (DOUBLE BYTE ADD)
I-DBF A5 72	LDA	COLAC	ADD
I-DC1 65 77	ADC	DELTAC	
I-DC3 05 72	STA	COLAC	
I-DC5 A5 73	LDA	COLAC+1	
I-DC7 65 78	ADC	DELTAC+1	
I-DC9 05 73	STA	COLAC+1	
I-DCB C5 75	CHP	ENDPT+1	COMPARE HIGH BYTE
I-DCD 90 27	BCC	DRAWB	
I-DCF D0 06	BNE	DRAW6A	

I-DD1 A5 72		LDA	COLAC	COMPARE LOW BYTE
I-DD3 C5 74		CHP	ENDPT	
I-DD5 90 1F		RCC	DRAWB	
I-DD7 24 7A	DRAW6A	BIT	COLINC	+ OR - ?
I-DD9 10 10		BPL	DRAW6B	
I-DDB C6 55		DEC	COLCRS	DO DOUBLE BYTE DECREMENT
I-DDD A5 55		LDA	COLCRS	
I-DDF C9 FF		CHP	#\$FF	
I-DE1 D0 0E		BNE	DRAW7	
I-DE3 A5 56		LDA	COLCRS+1	
I-DE5 F0 0A		BEQ	DRAW7	DONT DEC IF ZERO
I-DE7 C6 56		DEC	COLCRS+1	
I-DE9 10 04		BPL	DRAW7	(UNCONDITIONAL)
I-DEB E6 55	DRAW6B	INC	COLCRS	DO DOUBLE BYTE INCREMENT
I-DED D0 02		BNE	DRAW7	
I-DEF E6 56		INC	COLCRS+1	
I-DF1 A2 02	DRAW7	LDX	#2	AND SUBTRACT ENDPT FROM COLAC
I-DF3 20 7A FA		JBR	BUBEND	
I-DF6 20 76 FA	DRAWB	JBR	RANGE	
I-DF9 20 E0 F5		JBR	OUTPLT	PLOT POINT
I-DFC AD D7 02		LDA	FILFLO	TEST RIGHT FILL
I-DFE F0 2F		BEQ	DRAW9	
I-E01 20 9D FC		JSR	PHACRS	
I-E04 AD FB 02		LDA	ATACHR	
I-E07 BD BC 02		STA	HOLD4	
I-E0A A5 B4	DRAWBA	LDA	ROWCRS	SAVE ROW IN CASE OF CR
I-E0C 48		PIA		
I-E0D 20 DC F9		JBR	INCRSA	POSITION CURSOR ONE PAST DOT
I-E10 68		PLA		RESTORE ROWCRS
I-E11 B5 54		STA	ROWCRS	
I-E13 20 76 FA	DRAWB	JBR	RANGE	
I-E16 20 A2 F5		JBR	OUTPLT	GET DATA
I-E19 D0 0C		BNE	DRAWBB	STOP IF NON-ZERO DATA IS ENCOUNTERED
I-E1B AD FD 02		LDA	FILDAT	FILL DATA
I-E1E BD FB 02		STA	ATACHR	
I-E21 20 E0 F5		JBR	OUTPLT	DRAW LOOP
I-E24 4C 0A FE		JMP	DRAWBA	
I-E27 AD BC 02	DRAWBB	LDA	HOLD4	
I-E2A BD FB 02		STA	ATACHR	
I-E2D 20 AB FC		JSR	PLACRS	
I-E30 38	DRAW9	BEC		DO DOUBLE BYTE SUBTRACT
I-E31 A5 7E		LDA	COUNTR	
I-E33 E9 01		BBC	#1	
I-E35 B5 7E		STA	COUNTR	
I-E37 A5 7F		LDA	COUNTR+1	
I-E39 E9 00		BBC	#0	
I-E3B B5 7F		STA	COUNTR+1	
I-E3D 00 03		BMI	DRAW10	
I-E3F 40 90 FD		JMP	DRAW4A	
I-E42 4C 34 F6	DRAW10	JMP	RETURN	

TABLES

MEMORY ALLOCATION

FE4B 1B 10 0A ALOCAT BYTE 24,16,10,10,16,28,52,100,196,196,196,196
 FE4B 0A 10 1C
 FE4B 34 64 C4
 FE4E C4 C4 C4

NUMBER OF DISPLAY LIST ENTRIES

FE51 17 17 0B NUMDLE BYTE 23,23,11,23,47,47,95,95,97,97,97,97
 FE54 17 2F 2F
 FE57 5F 5F 61
 FE5A 61 61 61
 FE5D 13 13 09 NXMDE BYTE 19,19,9,19,39,39,79,79,65,65,65,65 (EXTENSION OF NUMDLE)
 FE60 13 27 27
 FE63 4F 4F 41
 FE66 41 41 41

ANTIC CODE FROM INTERNAL MODE CONVERSION TABLE

INTERNAL	ANTIC CODE	DESCRIPTION
0	2	40X2XB CHARACTERS
1	6	20X5XB "
2	7	20X5X1A "
3	8	40X4XB GRAPHICS
4	9	80X2X4 "
5	A	80X4X4 "
6	B	160X2X2 "
7	D	160X4X2 "
8	F	320X2X1 "
9		SAME AS B BUT Q1A 'LUM' MODE
10		SAME AS B BUT Q1A 'COL/LUM REGISTER' MODE
11		SAME AS B BUT Q1A 'COLOR' MODE

FE69 02 04 07 ANCONV BYTE 0,4,7,8,9,9A,9B,9D,9F,9F,9F,9F (ZEROS FOR RANGE TEST IN PAGETB)
 FE6C 08 09 0A
 FE6F 0B 0D 0F
 FE72 0F 0F 0F

PAGE TABLE TELLS WHICH DISPLAY LISTS ARE IN DANGER OF CROSSING A 256 BYTE PAGE BOUNDARY

FE75 00 00 00 PAGETB BYTE 0,0,0,0,0,0,0,1,1,1,1,1
 FE78 00 00 00
 FE7B 00 01 01
 FE7E 01 01 01

THIS IS THE NUMBER OF LEFT SHIFTS NEEDED TO MULTIPLY COLCRS BY 10,20, OR 40. (ROWCRS*10)/(2*DHLNE)

FE81 02 01 01 DHLNE BYTE 2,1,1,0,0,1,1,2,2,2,2,2

IEB4 00 00 01
IEB7 01 02 02
IEB8 02 02 02

COLUMN: NUMBER OF COLUMNS

IEB8 28 14 14 COLUMN BYTE 40, 20, 20, 40, 80, 80, 160, 160, 64, 80, 80, 80 MODE 8 IS SPECIAL CASE
IE90 28 50 50
IE93 A0 A0 40
IE96 50 50 50

NOROWS: NUMBER OF ROWS

IE99 18 18 0C NOROWS BYTE 24, 24, 12, 24, 48, 48, 96, 96, 192, 192, 192, 192
IE9C 18 30 30
IE9F 60 60 0C
IEA2 0C 0C 0C

DIV2TB: HOW MANY RIGHT SHIFTS FOR HCRB FOR PARTIAL BYTE MODES

IEA5 00 00 00 DIV2TB BYTE 0, 0, 0, 2, 3, 2, 3, 2, 3, 1, 1, 1
IEA8 02 03 02
IEA8 03 02 03
IEAE 01 01 01

DMASKT: DISPLAY MASK TABLE

IEB1 00 FF F0 DMASKT BYTE \$00, \$FF, \$F0, \$0F
IEB4 0F
IEB5 00 30 0C
IEB8 03

MASKTB: BIT MASK. (ALSO PART OF DMASKTB! DO NOT SEPARATE)

IEB9 80 40 20 MASKTB BYTE \$80, \$40, \$20, \$10, \$08, \$04, \$02, \$01
IEBC 10 08 04
IEBF 02 01

COLRTB: BYTE \$28, \$CA, \$94, \$46, \$00

IEC1 28 CA 94
IEC4 46 00

CNTRLB: CONTROL CODES AND THEIR DISPLACEMENTS INTO THE CONTROL CHARACTER PROCESSORS

IEC6 1B CNTRLS BYTE \$1B
IEC7 79 F7 WORD ESCAPE
IEC9 1C BYTE \$1C

IECA 7F F7	WORD	CRGRUP
IECC 1D	BYTE	\$1D
IECD 8C F7	WORD	CRSRDN
IECF 1E	BYTE	\$1E
IED0 99 F7	WORD	CRSLF
IED2 1F	BYTE	\$1F
IED3 AA F7	WORD	CRSRT
IED5 7D	BYTE	\$7D
IED6 89 F7	WORD	CLRSR
IED8 7E	BYTE	\$7E
IED9 E6 F7	WORD	BS
IEDB 7F	BYTE	\$7F
IEDC 10 F8	WORD	TAB
IEDF 9B	BYTE	\$9B
IEEF 30 FA	WORD	DOCRWS
IEE1 9C	BYTE	\$9C
IEE2 D4 F8	WORD	DELLIN
IEE4 9D	BYTE	\$9D
IEE5 A4 F8	WORD	INSLIN
IEE7 9E	BYTE	\$9E
IEE8 32 F8	WORD	CLRTAB
IEEA 9F	BYTE	\$9F
IEEB 2D F8	WORD	SETTAB
IEED FD	BYTE	\$FD
IEEE 0A F9	WORD	BELL
IEF0 FE	BYTE	\$FE
IEF1 6D F8	WORD	DELCHR
IEF3 FF	BYTE	\$FF
IEF4 37 F8	WORD	INSCHR

ATAINT: ATASCII TO INTERNAL TABLE

IEF6 40 00 20	ATAINT	BYTE	\$40, \$00, \$20, \$40
IEF9 60			

INTATA: INTERNAL TO ATASCII TABLE

IEFA 20 40 00	INTATA	BYTE	\$20, \$40, \$00, \$40
IEFD 60			

ATASCI: ATASCII CONVERSION TABLE

IEFE 6C 6A 0B	ATASCI	BYTE	\$6C, \$6A, \$3B, \$80, \$80, \$6B, \$2B, \$2A LOWER CASE
IF01 80 80 6B			
IF04 2B 2A			
IF06 6F 80 70	BYTE		\$6F, \$80, \$70, \$75, \$7B, \$69, \$2D, \$3D
IF09 75 9B 69			
FF0C 2D 3D			
FF0E 76 80 63	BYTE		\$76, \$80, \$63, \$80, \$80, \$62, \$7B, \$7A
FF11 80 80 62			
IF14 7B 7A			
IF16 34 80 33	BYTE		\$34, \$80, \$33, \$36, \$1B, \$35, \$32, \$31
FF19 36 1B 39			

IF1C 32 31

IF1E 2C 20 2E . BYTE \$2C, \$20, \$2E, \$6E, \$80, \$6D, \$2F, \$81

IF21 6E 80 6D

IF24 2F 81

IF26 72 80 65 . BYTE \$72, \$80, \$65, \$79, \$7F, \$74, \$77, \$71

IF29 79 7F 74

IF2C 77 71

IF2E 39 80 30 . BYTE \$39, \$80, \$30, \$37, \$7E, \$38, \$3C, \$3E

IF31 37 7E 38

IF34 3C 3E

IF36 66 68 64 . BYTE \$66, \$68, \$64, \$80, \$82, \$67, \$73, \$61

IF39 80 82 67

IF3C 73 61

IF3E 40 4A 3A . BYTE \$4C, \$4A, \$3A, \$80, \$80, \$4B, \$50, \$5E UPPER CASE

IF41 80 80 4B

IF44 5C 5E

IF46 4F 80 50 . BYTE \$4F, \$80, \$50, \$55, \$9B, \$49, \$5F, \$7C

IF49 55 9B 49

IF4C 5F 7C

IF4E 56 80 43 . BYTE \$56, \$80, \$43, \$80, \$80, \$42, \$5B, \$5A

IF51 80 80 42

IF54 5B 5A

IF56 24 80 23 . BYTE \$24, \$80, \$23, \$26, \$1B, \$25, \$22, \$21

IF59 26 1B 25

IF5C 22 21

IF5E 5B 20 5D . BYTE \$5B, \$20, \$5D, \$4E, \$80, \$4D, \$3F, \$81

IF61 4E 80 4D

IF64 3F 81

IF66 52 80 45 . BYTE \$52, \$80, \$45, \$59, \$9F, \$54, \$57, \$51

IF69 59 9F 54

IF6C 57 51

IF6E 2B 80 29 . BYTE \$2B, \$80, \$29, \$27, \$9C, \$40, \$7D, \$9D

IF71 27 9C 40

IF74 7D 9D

IF76 46 4B 44 . BYTE \$46, \$4B, \$44, \$80, \$80, \$47, \$43, \$41

IF79 80 83 47

IF7C 53 41

IF7E 0C 0A 7B . BYTE \$0C, \$0A, \$7B, \$80, \$80, \$0B, \$1E, \$1F CONTROL

IF81 80 80 0B

IF84 1E 1F

IF86 0F 80 10 . BYTE \$0F, \$80, \$10, \$15, \$9B, \$09, \$1C, \$1D

IF89 15 9B 09

IF8C 1C 1D

IF8E 16 80 03 . BYTE \$16, \$80, \$03, \$80, \$80, \$02, \$1B, \$1A

IF91 80 80 02

IF94 1B 1A

IF96 80 80 85 . BYTE \$80, \$80, \$85, \$80, \$1B, \$80, \$FD, \$80

IF99 80 1B 80

IF9C FD 80

1FAE 80 80 80	BYTE	\$80, \$80, \$80, \$80, \$FE, \$80, \$7D, \$FF
1FB1 80 FE 80		
1FB4 7D FF		
1FB6 06 0B 04	BYTE	\$06, \$0B, \$04, \$80, \$84, \$07, \$13, \$01
1FB9 80 84 07		
1FBC 13 01		

```

FFF2 FF FF FF . BYTE 0FF,0FF,0FF,0FF,0FF,0FF
FFF5 FF FF FF

```

0014 00 KBDSPR BYTE *FFFB-CRNTPC DISPLC IS TOO LONG
END

SYMBOL TABLE							
ADDCCR	030E	ADRESB	0064	AFP	D800	ALLPOT	D208
ADICAT	FE45	ANCONV	FE69	ANTIC	D400	APPEND	0001
ADIMU	000E	ASCCD1	F705	ATACHR	02FB	ATAINT	FEF6
ALAN	DE43	ATASCI	FEFE	ATTRACT	004D	AUDC1	D201
AUDC2	D203	AUDC3	D205	AUDC4	D207	AUDCTL	D208
AUDC1	D200	AUDF2	D202	AUDF3	D204	AUDF4	D206
BADMOD	0086	BADMOD	0091	BELL	F90A	BELL1	F90C
BFENLO	0035	BFENLO	0034	BITCLR	FB12	BITCON	FAEB
BITGET	FB31	BITGET	FB25	BITHSK	006E	BITPUT	FB04
BITSET	FAFA	BITSET	FD06	BLIM	029A	BLKBDV	E471
BOOTAB	0009	BOOTAB	0242	BOTSCR	02BF	BPTR	003D
BRKKEY	0080	BRKKEY	0011	BS	F7E6	BS1	F80D
BS	F805	BS3	F7E3	BSS	F7EC	BUFADR	0015
BUFRFL	006B	BUFRFL	0038	BUFRHI	0033	BUFRLO	0032
CASBUF	006C	CASBUF	03FD	CASETV	E440	CASFLQ	030F
CASORQ	0002	CASORQ	EF41	CASSET	004B	CASSET	0043
CAUX1	023C	CAUX2	023D	CBAUDH	02EF	CBAUDL	02EE
CDEVID	023B	CDEVID	023A	CDTHA1	0226	CDTHA2	022B
CDTHF3	022A	CDTHF4	022C	CDTHF5	022E	CDTHV1	021B
CDTHV3	021A	CDTHV3	021C	CDTHV4	021E	CDTHV5	0220
CH	02FC	CH1	02F2	CHACT	02F3	CHACTL	D401
CHAB	02FA	CHBAS	02F4	CHBASE	D409	CHKERR	008F
CHKSUM	0038	CHKSUM	0031	CHROQ	E000	CIOCHR	002F
CIOINV	E46E	CIOORQ	E4A6	CIOV	E456	CIX	00F2
CLICK	004A	CLICK	FCDB	CLICK1	FCDA	CLOBE	000C
CLRCID	007D	CLRL11	FBA6	CLRLIN	F89B	CLR8C2	F7BF
CLRCR	F7CE	CLR8CR	F7B9	CLRTAB	F832	CLRTB8	F430
CNTRL	009F	CNTRL8	FEC6	COLAC	0072	COLBK	D01A
COLCR	FCE4	COLCR1	FCEE	COLCR2	FCF0	COLCR8	0055
COLDBT	0244	COLDBV	E477	COLINO	007A	COLORO	02C4
COLDR1	02C5	COLDR2	02C6	COLDR3	02C7	COLDR4	02C8
COLPF0	D016	COLPF1	D017	COLPF2	D018	COLPF3	D019
COLPH0	D012	COLPH1	D013	COLPH2	D014	COLPH3	D015
COLRTB	004F	COLRTB	FEC1	COLUMN	FEBD	COMRE1	F7A7
CONRET	F7B9	CONROL	D01F	CONVR1	F97E	CONVR2	F98B
CONVR3	F98F	CONVR4	F99C	CONVR5	F9A6	CONVR6	F9A7
CONVRT	F947	COS	BD73	COUNTR	007E	CR	0098
CRITRY	0036	CRITIC	0042	CRLOOP	F5E7	CRNTPC	FFF8
CRSTNI	02F0	CRSRDN	F78C	CRSRL1	F7A3	CRSRLF	F799
CRSROR	008D	CRSRRT	F7AA	CRSRUP	F77F	CROPIV	E47D
CSSTAT	0288	CTIA	D000	DAUX1	030A	DAUX2	030B
DBDEC	F913	DBDEC	F91F	DBRECT	0241	DBBUB	F921
DBSUB1	F934	DBUFH1	0305	DBUFLO	0304	DBYTH1	0309
DBYTH0	0308	DCB	0300	DCOMND	0302	DDEVIC	0300
DELFLQ	00F8	DEGON	0006	DELCH1	F870	DELCH2	F896
DELCIR	F86D	DELETE	0021	DELL11	F8DD	DELL12	F8FB
DELL1A	F8D7	DELL1B	F8DB	DELL1N	F8D4	DELTAC	0077
DELTAR	0076	DELT11	FC82	DELT12	FC89	DELT13	FCBC
DELTIA	FC68	DELT1B	FC70	DELT1M	FC73	DERROR	0090
DLACB	0240	DHLINE	FEB1	DIGRT	00F1	DINDEX	0057
DIRECT	0002	DISK	0044	DISKIV	E450	DISPLA	E410
DISPLY	0053	DIV2TB	FEA5	DLISTH	D403	DLISTL	D402
DMACTL	D40Q	DMASK	02A0	DMASKT	FEB1	DNACK	008B
DMU1	FC1A	DOBU1A	FC29	DOBUF1	FC12	DOBUF2	FC39
DOBUF3	FC51	DOBUF4	FC55	DOBUFC	FC00	DOCR	FA34
DOCR1	FA00	DOCR1A	FA29	DOCR1B	FA14	DOCR2	FA3D
DOCR2A	FA4A	DOCR2B	FA4D	DOCR4B	FA61	DOCRWB	FA30
DOLC01	FBE5	DOLC02	FBFB	DOLCOL	F8DD	DOPEN	F3F6
DOPEN1	F460	DOPEN2	F4D5	DOPEN3	F4FA	DOPEN4	F51C

DOPEN5	F524	DOPEN7	F588	DOPEN8	F438	DOPEN9	F577
DOPENA	F457	DOSINI	000C	DQSS	F6AD	DQVEEC	000A
DRAW	FCFC	DRAW1	FD37	DRAW10	FE42	DRAW11	FD99
DRAW*	FD5C	DRAW3	FDB3	DRAW3A	FD62	DRAW4A	FD90
DRAW*	FDA4	DRAW5A	FDB2	DRAW6	FDBE	DRAW6A	FDD7
DRAWB	FDEB	DRAW7	FDF1	DRAW8	FDF6	DRAW8A	FE0A
DRAWB	FE27	DRAWBC	FE13	DRAW9	FE30	DRAWA	FDOC
DRAWL	FDO8	DRAWLN	Q011	DRETRY	Q037	DRKMSK	Q04E
DSKIN	001B	DSKINV	E453	DSKORG	EDEA	DSKTIM	Q246
DSKIN	001A	DSPFLQ	Q2FE	DSTAT	Q04C	DSTATS	Q303
DLINQ	Q306	DUNIT	Q301	DUNUSE	Q307	DVSTAT	Q2FA
EDITR	E400	EDITRV	E400	EEXP	Q0ED	EOETC1	F450
EOETC2	F46E	EOETC3	F47C	EOETC5	F491	EOETC6	F46B
EOETC7	F480	EOETCH	F43E	ENDPT	Q074	EOERR	Q08B
EOUT	F3FC	EOUTC5	F4BE	EOUTC6	F4B5	EOUTCH	F4A4
ERANCE	FAB8	ERETN	F4BB	ERRFLQ	Q23F	ESCAPE	F779
ESCLL	Q2A2	ESIGN	Q0FF	EXP	DDC0	EXP10	DDCC
EXTEN1	F87E	EXTEN3	F8BB	EXTEN4	F892	EXTEND	F87B
FADP	DA66	FASC	D8E6	FCHRFL	Q0F0	FDIV	DB2B
FLUE	Q03E	FILDAT	Q2ED	FILELQ	Q2B7	FILLIN	Q012
FLDOR	DD8D	FLDOR	DD89	FLDIP	DD9C	FLDIR	DD9B
FLPTR	Q0FC	FMOVE	DD86	FMSZP0	Q043	FMUL	DADB
FNCHIT	Q092	FORMAT	Q022	FPI	D9D2	FPREC	Q006
FPCR	Q5E6	FPBCR1	Q5EC	FPTR2	Q0FE	FRO	Q0D4
FRI	Q0E0	FR2	Q0E6	FRE	Q0DA	FREQ	Q040
FRMADR	Q069	FRMERR	Q08C	FRX	Q0EC	FRCH	Q5E6
FGRU	Q5EC	FSTOP	DDAB	FSTOR	DDA7	FSUB	DA60
FTYPE	Q03E	GETCH	F593	GETCHR	Q007	GETOUT	F749
GLTIL	F5A2	GETREC	Q005	GLBABS	Q2E0	GPRIOR	Q26F
GRAC1	D01D	GRAFM	D011	GRAFP0	D00D	GRAFP1	D00E
GRAFP2	D00F	GRAFP3	D010	HATAB8	Q31A	HITCLR	D01E
HOLD1	Q051	HOLD2	Q29F	HOLD3	Q29D	HOLD4	Q28C
HOLD5	Q2BD	HOLDCH	Q07C	HOME	F7D6	HPOSM0	D004
HPOSM1	D005	HPOSM2	D006	HPOSM3	D007	HPOBP0	D000
HPOSM4	D001	HPOSP2	D002	HPOSP3	D003	HSCROL	D404
ICAX1	Q34A	ICAX1Z	Q02A	ICAX2	Q34B	ICAX2Z	Q02B
ICBA1	Q345	ICBAH2	Q025	ICBAL	Q344	ICBALZ	Q024
ICBL1	Q349	ICBLH2	Q029	ICBL	Q34B	ICBLZ	Q02B
ICCH	Q342	ICCOMT	Q017	ICCOMZ	Q022	ICDND	Q341
ICDNDZ	Q021	ICHID	Q340	ICHIDZ	Q020	ICIDND	Q02E
ICPT1	Q347	ICPTHZ	Q027	ICPTL	Q346	ICPTLZ	Q026
ICPR	Q34C	ICSPRZ	Q02C	ICSTA	Q343	ICSTA2	Q023
INP	D9AA	INATA1	F84A	INATAC	F832	INBUFF	Q0F3
INCPA	F9FB	INCRB1	FA77	INCRB2	F9E4	INCRB3	F9F7
INCRBA	F9DC	INCRSB	F9D4	INCRSC	F9DA	INCRSR	F9DB
INSC1	F852	INSC3	F85E	INSC4	F844	INSC5	F86A
INSC16	F861	INSCR	F837	INSLR	Q020	INSDAT	Q07D
INSL1	F8C6	INSL12	F8CE	INSL1A	F8A5	INSLIN	F8A4
INTAB	Q200	INTATA	FEFA	INTEMP	Q22D	INTINV	E46B
INTLRO	E6D5	INTZBB	Q010	INVFLQ	Q2B6	IOCB	Q340
IOCBAS	Q020	IOCB6Z	Q010	IOCFRE	Q0FF	IRGEN	D20E
IRGT	D20E	JBRIND	F6A1	K1	F729	K2	F734
K1	F73F	K4	F776	K5	F76B	K6	F74D
K7	F745	K8	F773	KBCDE	D209	KBD	Q04B
KDDIND	E420	KBDDOR	F3E4	KBDSPR	Q014	KEYBDV	E420
KEYDEL	Q2F1	KOETC1	F71E	KOETC2	F6DD	KOETC3	F6FE
KCFICH	F6E2	LBFEND	Q5FF	LBPR1	Q57E	LBPR2	Q57F
LBUFF	Q580	LEDGE	Q002	LFRTCH	F7A5	LINBUF	Q247
LINZIB	Q000	LMARGN	Q052	LO1GET	F822	LO2GET	F823
LOGKIL	Q023	LOG	DECD	LOG10	DED1	LOGCIL	Q063

LOGM1	FB20	LOGMAP	02B2	LPENH	0234	LPENV	0235
NOPI	D000	MOPL	D00B	MIPF	D001	M1PL	D009
NETT	D002	M2PL	D00A	M3PF	D003	M3PL	D00B
NASKID	FEB9	MAXDEV	0021	MAXIOC	0080	MEMLO	02E7
NMTMP	02E5	MLTMP	0066	MODEM	004D	MONORG	F0E3
NOM11	F85B	MOVL12	FB7A	NOVLIN	FB4E	MXDMDE	FE5D
NXDMID	0010	NEWCOL	0061	NEWROW	0060	NMIEN	D40E
NMTMB	D40F	NMIST	D40F	NCKGM	003C	NOFUNC	F63D
NORID	F4AB	NONDEV	00B2	NOROWB	FE99	NOBCRI	FA32
NOSID	FA2C	NOTB	F4BB	NOTE	0026	NOTMXD	F4F5
NOTIBN	00B5	NSIGN	00EE	NUNDLE	FE51	NVALID	00B4
OFFERR	FAE4	OLDADR	005E	OLDCHR	005D	OLDCL	005B
OLDRAW	005A	OPEN	0003	OPNCOM	F404	OPNERR	F453
OPNIN	0004	OPNIND	000C	OPNOT	000B	OPNTMP	0066
OUTCH	F5B7	OUTCH2	F5FF	OUTCHA	F5BD	OUTCHB	F5D7
OUTCHF	F5CA	OUTPLT	F5E0	OVRRUN	00BE	POPF	D004
P0M	D00C	P1PF	D005	P1PL	D00D	P2PF	D006
P2M	D00E	P3PF	D007	P3PL	D00F	PACTL	D302
PADD10	0270	PADDL1	0271	PADDL2	0272	PADDL3	0273
PADD14	0274	PADDL5	0275	PADDL6	0276	PADDL7	0277
PACTH	FE75	PBCTL	D303	PBPNT	001D	PBUFSZ	001E
PCDIR0	02C0	PCOLR1	02C1	PCOLR2	02C2	PCOLR3	02C3
PI-MI	D40C	PENV	D40D	PHACR1	FC9F	PHACRB	FC9D
PIA	D300	PIRG	FFBE	PIRG1	FFDC	PIRG2	FFF0
PIRG3	FFCB	PIRG4	FFEB	PLACR1	FCAA	PLACRB	FCAB
PLYARG	05E0	PLYEVL	DD40	PMBASE	D407	POINT	0025
POKEY	D200	POKMSK	0010	PORTA	D300	PORTB	D301
POLD	D200	POT1	D201	POT2	D202	POT3	D203
POT4	D204	POT5	D205	POT6	D206	POT7	D207
POTGO	D20B	PRINTR	0050	PRINTV	E450	PRIOR	D01B
PINDRE	03C0	PRNDRO	EE7B	PRVOPN	00B1	PTMP	001F
PIMIT	001C	PTRIG0	027C	PTRIG1	027D	PTRIG2	027E
PTRIG3	027F	PTRIG4	0280	PTRIG5	0281	PTRIG6	0282
PTRIG7	0283	PUTCHR	000B	PUTMSC	FCF3	PUTREC	0009
PWINDA	F3E4	RADFL0	00FB	RADON	0000	RAMLO	0004
RANG12	02E4	RAMTOP	006A	RANDON	D20A	RANGE	FA96
RANGE1	FAB7	RANGE2	FAB8	RANGE3	FA9E	RBL0KV	E47A
RDINLY	00B7	RECVDN	0039	REDGE	0027	RENAME	0020
RETRU1	F634	RETRU3	FAE1	RETURN	F621	RHARON	0053
RNGR1	FAD8	RNGER2	FAD6	RNGERR	FAD1	RNGOK	FAC4
ROWAC	0070	ROWCRS	0054	ROWINC	0079	RTCLOK	0012
SAVADR	006B	SAVID	0316	SAVMBC	005B	SCREDT	0045
SCRENV	E410	SCRFLO	02B8	SCRHEM	0093	SCROL1	FB87
SCROL2	FBCA	SCROLL	FBAC	SDLSTH	0231	SDLSTL	0230
SDNCTL	022F	SENDEV	E46B	SERIN	D20D	SEROUT	D20D
SETTAD	FB2D	SETV8V	E45C	SHFANT	006F	SHFLOK	02BE
SHIT11	F5B1	SHIFT2	F610	SHIFTD	F5AA	SHIFTU	F60B
SHN	BD81	SIOTNV	E465	SIDORO	E944	SIOV	E459
SIZEN	D00C	SIZEPO	D00B	SIZEP1	D009	SIZEP2	D00A
SIZEP3	D00B	SKCTL	D20F	SKRES	D20A	SKSTAT	D20F
SOUND	0041	SPECIL	000E	SQR	BE81	SRTMR	022B
SW100	02FF	SSKCTL	0232	STACKP	031B	STAT1B	000D
STATUB	0030	STICK0	027B	STICK1	0279	STICK2	027A
STICK3	027B	STIMER	D209	STORE	F917	STORE1	F91D
STRID0	FC5C	STRERR	F942	STR100	02B4	STR1Q1	02B5
STR1Q2	02B6	STR1Q3	02B7	STROK	F946	SUBEND	FA7A
SUBTMP	029E	SUCCESS	0001	SWAP	FCB3	SWAP1	FCC2
SWAP1	FCD7	SWAPA	FCB9	SWPFL0	007B	SYSBVV	E45F
TAD	FB10	TAB1	FB23	TAB2	FB2A	TABMAP	02A3
TEMP	023E	TEMP1	0312	TEMP2	0314	TEMP3	0315

DISPLAY HANDLER -- 10-30-78 -- DISPLC

TIMER1	030C	TIMER2	0310	TIMFLO	0317	TIMOUT	008A
TINDEX	0293	TMPCHR	0050	TMPCOL	0289	TMPLBT	02A1
TMPROW	02B8	TMPX1	029C	TOADR	0066	TRAMSZ	0006
TRIG0	D010	TRIG1	D011	TRIG2	D012	TRIG3	D013
TRIG4	0089	TSTAT	0319	TSTCT1	FC8F	TSTCT2	FC9C
TSTCT1	FC8D	TSTDAT	0007	TXTCOL	0291	TXTMSC	0294
TXTCOL	0296	TXTROW	0290	UNLOCK	0024	UPDNCH	F7B7
VBANKA	0480	VBREAK	0206	VBWAIT	E496	VCRUNT	D40B
VCTANI	E480	VDELAY	D01C	VDSLST	0200	VECTDL	E400
VTHICK	0216	VINTER	0204	VKEYBD	0208	VPRCED	0202
VZCRCH	D405	VSERIN	020A	VSERQC	020E	VSERDR	020C
VTIMER1	0210	VTIMER2	0212	VTIMER4	0214	VVBLKD	0224
VVBLKI	0222	WARMST	000B	WARMBV	E474	WMODE	02B9
WARMY	00B3	WSYNC	D40A	XITVBV	E462	XMTDQN	003A
ZIOCH	0020	ZTEMP1	00F5	ZTEMP3	00F9	ZTEMP4	00F7

1417

Report Generated From: Inc. 10